PRIJS ƒ 7.95/Bfr. 160

Commodore INFO

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 6, NO. 3, april/mei 1989



LISTINGS

- Floppy-Hoes C-64
- Propschrift C-64
- Reken-Motief C-64
- Tone Dial C-64
- Wisser C-64
- Ruimterace C-64
- Taxi C-128
- Patience C-128
- Form Amiga

Red Storm Rising

Graphics op de C-64

GeoChart

Boot 128

CAD op de 128

Zany Golf Amiga

Amiga Zelfbouw: Spectrum-analyser

Vaste rubrieken

- Oud van Goudriaan
- Basic Miniatuurtjes
- Geos Info
- Amiga Machinetaal
- Tips & Trucs

Commodore Info
Verschijnt 8x per jaar
Jaarg.6, no.3, april/mei 1989

Uitgave:
Sala Communications
Uitgever:
Vic Sharfman
Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdred.
drs. J. Boers eindredacteur
drs. M. de Rooij &
J. Broekhuizen productie
drs. H. Zoete, H. Smeenk, drs. U. Schuurmans, R. Goudriaan, B. Munniksma, B. Venema, P. Boncz, MGCC/Johan & Johan

Redactiesecretariaat:
R. van Zalingen

Strip: Bert Tier
Illustraties: Ben van Mierlo

Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198
Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871
Listingtelefoon:
(ma: 17.00-21.00) 02155-25162

Abonnementen en administratie:
Nicole Balke en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.
Abonnement:
Voor 8 nummers *f* 47,50 of Bfr. 975 per jaar. Betaling op giro 1585491 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Omslagfoto: Homesoft Benelux 023-311241
Druk: NDB, Zoeterwoude
Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

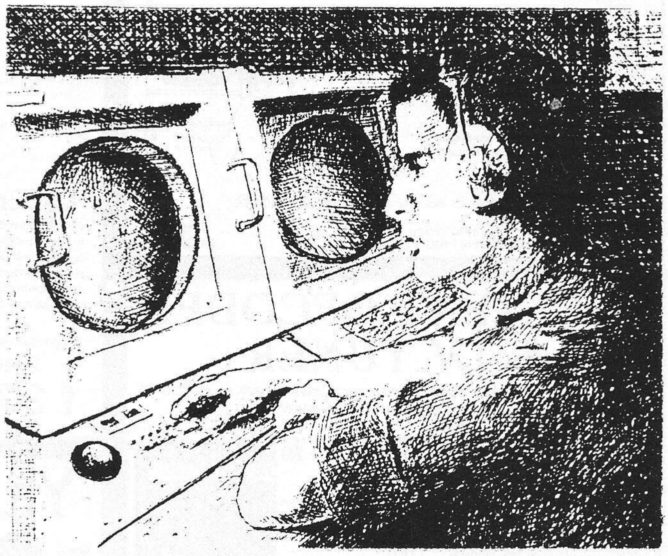
Inhoud van dit nummer

Redactioneel
Het lijkt erop, dat Commodore zijn traditionele januskop nog niet heeft afgezet. Aan de ene kant zegt het bedrijf zich vooral te richten op de professionele toepassingen, met de zwaardere PC's en Amiga modellen, waarvoor men fraaie concepten lanceert. Aan de andere kant moet natuurlijk wel blijven verkopen, en het merendeel van de omzet komt nog steeds van de trouwe C-64 en de goedkopere huiscomputermodellen. We hadden even het idee, dat men de PC's dan echt als zakelijke modellen zou blijven houden, daarmee boekt men in Duitsland veel succes en is Commodore bijna marktleider. Toen zette Kwantumhallen echter de PC-1 en de Amerikaanse Colt modellen, die toch typisch in de categorie budget-computers vallen, weer paginagroot in de kranten.
Commodore verkoopt bijzonder goed met die produkten en voor minder dan 1000 gulden een echte PC met 512 KB en alles wat er bij hoort, dat is ook een koopje. Het probleem is echter, dat op lange termijn de koper dan wel de naam Commodore blijft verbinden aan dergelijke koopjes. Dan is het moeilijker om doordachte, maar ook duurdere netwerk en multi-user systemen in de markt te zetten. Gelukkig hoeven de meeste lezers van dit blad zich daar verder geen zorgen over te maken, de 64 blijft, ondanks alle geruchten, nog steeds goed lopen en Commodore gaat er nog lang niet mee ophouden. Ook de Amiga doet het weer goed, we hebben voor het volgende nummer een hele reeks nieuwe spelsoftware van Mirrorsoft binnengekregen, er blijft genoeg nieuws komen.

Luc Sala

Red de wereld van de Sovjet overheersing, met behulp van de nieuwste onderzeeboot-simulator.....

Red Storm Rising



Red Storm Rising is een hart-kloppend simulatieprogramma waarbij de speler het commando heeft over een Amerikaanse nucleaire onderzeeboot. Met de Russen, die voorbereidingen treffen om de olievelden in het Midden Oosten over te nemen, komt het tot een botsing. De speler heeft hierbij een cruciaal aandeel in deze oorlog op zee, vechtend tussen de Russische beer en een totale overheersing van de wereld. Een combinatie van briljant strategisch- en tactisch denkwerk zijn noodzakelijk om deze technologische oorlogsvoering onder water te overleven. Het spel bevat trainingscenes, om mee te beginnen, gevolgd door opwindende zeeslagen van toenemende moeilijkheidsgraad. De uiterste uitdaging is de uiteindelijke zeeslag. Hierbij ervaart de speler het gehele verloop van de derde Wereldoorlog, zoals de schrijver, Tom Clancy, het op zo'n dramatische wijze heeft beschreven in zijn boek Red Storm Rising, de bestseller over moderne oorlogsvoering. Wat zal de uitkomst zijn in de strijd, met jou als commandant van een Amerikaanse onderzeeboot?

Het spel wordt geleverd voor verschillende computers, waaronder de C64. Het pakket bestaat uit een dubbelzijdige diskette, een toetsenbord-overlay, een Engelse handleiding en een technisch supplement met specifieke aanwijzingen per computertype. Het technisch supplement geeft de betekenis weer van specifieke toetsen en andere controletoetsen, die worden gebruikt bij de besturing van de onderzeeboot. De toetsenbord-overlay biedt een duidelijk overzicht van de functies welke een toets bezit. Wanneer het spel voor het eerst wordt gespeeld, luidt het advies een training te volgen, alvorens de veel moeilijker "battles" te spelen. Hierdoor krijg je ervaring met het gebruik van de toetsen. Met name de navigatie, het gebruik van de wapensystemen, sonar en radar vergen wel enige ervaring. Wanneer je eenmaal door de training heen geworsteld bent, wordt het tijd voor een echte "battle". Red Storm Rising heeft veel opties. Om een selectie te maken wordt gebruik gemaakt van de joystick, muis of cursortoetsen. Deze wordt in het spel de **controller** genoemd, terwijl de vuurknop of returntoets wordt aangeduid als **selector**.

Start-opties.

Er kan een keuze worden gemaakt uit een viertal jaren, waarin de "battle" zal plaatsvinden. Afhankelijk van het gekozen jaar, beschikt men in het spel over geavanceerde bewapening. Maar de tegenpartij, de Russen, beschikken eveneens, afhankelijk van het gekozen jaar over modernere schepen en bewapening. Na deze keuze wordt een identificatietest afgenomen. Hierbij verschijnt een bepaald type oorlogsschip op het scherm, waarvan je de klasse correct moet intypen. Wanneer je een correct antwoord geeft mag je je naam intikken en kun je verder gaan. Zo niet, wordt je teruggestuurd naar de trainingschool. Vervolgens kan een keuze worden gemaakt met welk schip je de oorlog in wilt. Bepaalde types zijn, afhankelijk van het eerder gekozen jaartal wel of niet beschikbaar. Een regel hierbij is, hoe hoger het jaartal hoe nieuwer de schepen. Hoe nieuwer de schepen, des te stiller ze zijn en des te meer bewapening ze kunnen meevoeren. Zo zul je een schip uit de Seawolf-klasse, die uiterst krachtig is, pas kunnen kiezen wanneer je als jaartal 1996 hebt geselecteerd. Naast dit alles bestaat de mogelijkheid om een bepaald "level" te selecteren. Hierbij kan je kiezen uit een introductie-, een normale-, een serieuze- en uiteindelijke uitdaging. Het spel bevat drie scenario's, zoals training-acties, oorlogs-simulaties en

de uiteindelijk derde Wereldoorlog. De speler krijgt zijn orders van de computer via een satelliet. Deze kunnen gedurende het spel wijzigen. Het is mogelijk om gedurende het spel te pauzeren door op de P-toets te drukken. Dit geeft de speler de mogelijkheid even op adem te komen en het geheel te analyseren.

Toetsenbord gebruik.

Het gebruik van het toetsenbord in dit moeilijke spel is van groot belang, daar de meeste functies door een toetsaanslag worden gerealiseerd.

Zo kan met behulp van de F1-toets worden geschakeld tussen een tactische weergave of één van de andere informatieschermen, zoals sonar-vergelijking, wapencontrole, defensie-weergave, periscoop en scheeps-databank. Deze functies worden allemaal geactiveerd met behulp van de functietoetsen. Een bepaald scherm blijft steeds zichtbaar tot een andere toets wordt ingedrukt en daarmee de bijbehorende functie wordt geactiveerd. De snelheid waarmee de onderzeeboot vaart wordt geregeld met de + en - toetsen. De diepte wordt bepaald nadat men op de CLR/HOME toets heeft gedrukt, zodat de diepte ingevoerd kan worden door een driecijferig getal in te tikken. Koerswijzigingen worden ingesteld met behulp van de INST/DEL toets. Om een torpedo te laden gebruikt men de 4-toets en om projectielen af te vuren wordt gebruik gemaakt van de toetsen 5 t/m 8. Er is ook een help-toets, die je in staat stelt informatie op te vragen uit tactiek-computer.



Conclusie.

Red Storm Rising is één van de meest natuurgetrouwe simulatoren, waarover zeer goed is nagedacht. Het is niet een spel voor de speler die na het opstarten gelijk resultaat wil zien. Je moet er voor gaan zitten en bereid zijn eerst de handleiding grondig door te nemen, alvorens het geweld op het scherm los te laten barsten. Zeer geavanceerde, realistische bewapening, maar ook een zeer natuurgetrouwe navigatie-simulatie maken het spelen van dit spel voor de ware strateeg tot een plezier. Enkele karakteristieken van het pakket zijn o.m. zeer realistische simulatie van nucleaire onderzeeboten, intelligente computergestuurde wapenssystemen, een ontstellende variatie aan onderwatergebieden, een scala aan uitdagingen. Red Storm Rising is geprogrammeerd door de befaamde ontwerper/programmeur Sid Meier, die eerder al F-16 Strike Eagle en Silent Service heeft ontwikkeld. Het pakket is verkrijgbaar voor de C64 en C128 zowel op tape als op diskette en kost ongeveer fl. 79,00.

Bert Venema

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker dan de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

```
1       rem ********************************
        ***
  2     rem basic loader "SYNTAX.CHECKSUM"
  3     rem na de commando's "run" en "new
        "
  4     rem blijft dit programma in het ge
        -
  5     rem heugen. laad het te testen pro
        -
  6     rem gramma en tik daarna sys 49152
        .
  7     rem ********************************
        ***
10      i=49152   :rem beginadres
20      reada:ifa<0then40:rem data ingelez
        en
30      pokei,a:i=i+1:b=b+a:goto20
40      if b<>16844thenprint"[SHIFT-CLR]fo
        ut[SPACE]in[SPACE]dataregels!":b=0
        :end
50      poke49184,148:poke49185,192
55      i=49300
60      read a: ifa<0then80
70      pokei,a:b=a+b:i=i+1:goto60
80      if b<>20068thenprint"[SHIFT-CLR]fo
        ut[SPACE]in[SPACE]dataregels![SPAC
        E](vanaf[SPACE]regel[SPACE]240)":b
        =0:end
90      print"data[SPACE]is[SPACE]weggezet
        "
95      print"checksum[SPACE]testen[SPACE]
        met[SPACE]sys49152"
100     data 165,43,166,44,133,163,134,164
        ,169,147
110     data 32,210,255,160,0,240,3,32,73,
        192
120     data 32,73,192,208,1,96,32,225,255
        ,208
130     data 3,76,116,164,32,81,192,32,73,
        192
140     data 240,12,201,32,240,247,24,101,
        167,133
150     data 167,76,37,192,166,167,169,0,1
        32,168
160     data 32,205,189,169,13,32,210,255,
        164,168
170     data 76,17,192,200,208,2,230,164,1
        77,163
180     data 96,162,0,189,123,192,240,6,32
        ,210
190     data 255,232,208,245,32,73,192,170
        ,32,73
200     data 192,132,168,32,205,189,162,3,
        169,32
210     data 32,210,255,202,208,250,169,0,
        133,167
220     data 164,168,96,82,69,71,69,76,32,
        0
230     data  -1
240     data 165,197,201,3,240,7,201,4,240
250     data 6,76,148,192,76,34,192,169
260     data 147,32,210,255,76,161,192
270     data -1
```

** EINDE LISTING checksum 64 **

Checksum Checksum 64

REGEL 1	249	REGEL 110	158
REGEL 2	84	REGEL 120	232
REGEL 3	105	REGEL 130	183
REGEL 4	2	REGEL 140	96
REGEL 5	246	REGEL 150	96
REGEL 6	152	REGEL 160	127
REGEL 7	249	REGEL 170	71
REGEL 10	157	REGEL 180	223
REGEL 20	64	REGEL 190	73
REGEL 30	38	REGEL 200	79
REGEL 40	57	REGEL 210	109
REGEL 50	14	REGEL 220	106
REGEL 55	251	REGEL 230	225
REGEL 60	192	REGEL 240	16
REGEL 70	42	REGEL 250	163
REGEL 80	244	REGEL 260	92
REGEL 90	245	REGEL 270	22
REGEL 95	237		
REGEL 100	183		

PRINT OUT C-64 met o.a. Propschrift

Floppy-Hoes

Harold van Pinseren uit Boxtel heeft een programma gemaakt dat een directory van een schijf kan lezen en deze verkleint op een diskette hoes kan uitprinten. Het programma is geschreven voor de Star SG10. Maar het kan op een eenvoudige manier worden aangepast door de codes uit de eerste 18 regels te veranderen. Nalezen welke code uit uw handboek van de printer anders is, en deze aanpassen dan moet het programm goed kunnen werken. Heeft U een centronics-printer dan moet er eerst een extra opdracht worden gegeven. U typt dan in: sys2064:new

```
1     dimp$(17): rem *star sg-10*———
      ————————
2     p$(1)=chr$(27)+"8":        rem -ig
      nore paper-out signal
3     p$(2)=chr$(27)+"u"+chr$(0):rem -bi
      directional print
4     p$(3)=chr$(27)+"u"+chr$(1):rem -un
      idirectional print
5     p$(4)=chr$(27)+"a"+chr$(8):rem -se
      t line feed to 8/72 inch
6     p$(5)=chr$(27)+"e":        rem -em
      phasized print
7     p$(6)=chr$(27)+"g":        rem -do
      uble-strike print
8     p$(7)=chr$(27)+"f":        rem -ca
      ncel emphasized print
9     p$(8)=chr$(27)+"h":        rem -ca
      ncel double-strike print
10    p$(9)=chr$(27)+chr$(15):  rem -con
      densed print
11    p$(10)=chr$(27)+"3"+chr$(9):rem -s
      et line feed 9/144 inch
12    p$(11)=chr$(27)+"s"+chr$(0):rem -s
      uperscript on
13    p$(12)=chr$(18):           rem -pic
      a print
14    p$(13)=chr$(27)+"t":       rem -can
      cel superscript
15    p$(14)=chr$(27)+"2":       rem -set
       line feed to 1/6 inch
16    p$(15)=chr$(27)+"$"+chr$(1):rem -u
      se download characters
17    p$(16)=chr$(27)+"*"+chr$(0):rem -c
      opy rom to download ram
18    p$(17)=chr$(27)+"*"+chr$(1):rem -d
      efine download charac.
19    rem ————————————
20    rem ******************************
30    rem ********* floppy-hoes *********
40    rem ******************************
50    print"[SHIFT-CLR][COM-3][3xCRSR-DO
      WN][2xSPACE]zet[SPACE]de[SPACE]pri
      nter[SPACE]aan,"
60    print"[CRSR-DOWN][3xSPACE]en[SPACE
      ]druk[SPACE]<return>[2xCRSR-DOWN][
      CTRL-4]"
70    getq$:ifq$<>chr$(13)then70
80    print"[SHIFT-CLR][CTRL-4][11xCRSR-
      DOWN][6xSPACE]even[SPACE]geduld...."
90    gosub5030
100   rem ******************************
110   rem ********* floppy-hoes *********
120   rem ******************************
140   gosub6000
150   rem ————————————
200   dimna$(106),nb$(106),d$(145)
```

```
210   sp$="[69xSPACE]"
220   nn$(1)="[COM-A]CCCCCCCCCCCCCCCCC[
      COM-S]"
230   nn$(2)="B[SPACE]free[SPACE]blocks:
      [SPACE]...[SPACE]B"
240   nn$(3)="B[4xSPACE]programs:[SPACE]
      ...[SPACE]B"
250   nn$(4)="B[3xSPACE]disk[SPACE]lock:
      [SPACE]...[SPACE]B"
260   nn$(5)="[COM-Q]CCCCCCCCCCCCCCCCC[
      COM-W]"
270   nn$(6)="B[SPACE]>commodore-info<[S
      PACE]B"
280   nn$(7)="[COM-Z]CCCCCCCCCCCCCCCCC[
      COM-X]"
290   fori=100to106:nb$(i)=na$(i):next
300   rem ****************************
310   rem ****** begin programma ******
320   rem ****************************
330   poke53280,0:poke53281,0:rem poke65
      0,0:poke809,255
340   print"[CTRL-N][CTRL-H]"
350   print"[SHIFT-CLR][COM-6][CTRL-9][4
      0xSPACE]";
360   print"[3xSPACE]FLOPPY-HOES[4xSPACE
      ]voor[SPACE]de[2xSPACE]STAR[SPACE]
      SG-10[3xSPACE]";
370   print"[40xSPACE]";
380   print"[COM-3][CTRL-9][SPACE][38xCO
      M `][SPACE]";
390   print"[SPACE][COM-G][36xSPACE][COM
      M][SPACE]";
400   print"[SPACE][COM-G][6xSPACE]dit[S
      PACE]is[SPACE]een[SPACE]programma[
      SPACE]van[6xSPACE][COM-M][SPACE]";
410   print"[SPACE][COM-G][36xSPACE][COM
      M][SPACE]";
420   print"[SPACE][COM-G][11xSPACE]comm
      odore-info[11xSPACE][COM-M][SPACE]";
430   print"[SPACE][COM-G][36xSPACE][COM
      M][SPACE]";
440   print"[SPACE][38xCOM-T][SPACE][CTRL 8]"
450   print"[SPACE]Dit[SPACE]programma[S
      PACE]maakt[SPACE]een[SPACE]diskett
      e[SPACE]hoes"
460   print"[2xSPACE]waarop[SPACE]de[SPA
      CE]directory[SPACE]van[SPACE]de[SP
      ACE]diskette"
470   print"[4xSPACE]staat[SPACE]afgedru
      kt.[SPACE](DUBBELZIJDIG)"
480   print
490   print"————————
      ———";
500   print"[3xSPACE]1[SPACE]=[SPACE]all
      een[SPACE]hoes[SPACE]printout"
510   print"[CRSR-DOWN][3xSPACE]2[SPACE]
      =[SPACE]directory[SPACE]printout"
520   print"[CRSR-DOWN][3xSPACE]3[SPACE]
      =[SPACE]complete[SPACE]hoes[SPACE]
      printout"
530   print"[CRSR-DOWN][3xSPACE]D[SPACE]
      =[SPACE]directory"
540   print"————————
      ———";
550   print"[18xSPACE]MAAK[SPACE]EEN[SPA
      CE]KEUZE:[SPACE]";
560   getq$:ifq$=""then560
570   ifq$="1"thengosub700:goto300
580   ifq$="2"thengosub810:goto300
590   ifq$="3"thengosub910:goto300
```

```
600   ifq$="d"thengosub4000:goto300
610   goto560
700   rem ---------------------
710   tCTRL-8][2xCRSR-DOWN][4xSPAC
      E]veranderen[SPACE]?[2xSPACE](j/n)"
1$="B[SPACE]hoes[SPACE]printout[S
      PACE]B"
720   t2$="JCCCCCCCCCCCCCCCK"  :tt=11
730   gosub2600
740   print"[COM-3][2xCRSR-DOWN][2xSPACE
      ]zet[SPACE]het[SPACE]papier[SPACE]
      nu[SPACE]aan[SPACE]het[SPACE]begin."
750   print"[CRSR-DOWN][7xSPACE]en[SPACE
      ]druk[SPACE]<return>"
760   getq$:ifq$<>chr$(13)then760
770   print"[CTRL-8][3xCRSR-DOWN][6xSPAC
      E]printen[SPACE]hoes....."
780   gosub1000
790   return
800   rem ---------------------
810   t1$="B[SPACE]directory[SPACE]print
      out[SPACE]B"
820   t2$="JCCCCCCCCCCCCCCCCCCCCK":tt=9
830   gosub2600:gosub2000
840   gosub2600
850   print"[COM-3][3xCRSR-DOWN][2xSPACE
      ]zet[SPACE]het[SPACE]papier[SPACE]
      nu[SPACE]aan[SPACE]het[SPACE]begin,"
860   print"[CRSR-DOWN][7xSPACE]en[SPACE
      ]druk[SPACE]<return>"
870   getq$:ifq$<>chr$(13)then870
880   print"[CTRL-8][2xCRSR-DOWN][6xSPAC
      E]printen[SPACE]directory....."
890   gosub3000:return
900   rem ---------------------
910   t1$="B[SPACE]complete[SPACE]hoes[S
      PACE]printout[SPACE]B"
920   t2$="JCCCCCCCCCCCCCCCCCCCCCCCCK":tt=7
930   gosub730
940   gosub830
950   return
1000  rem ******************************
1010  rem ******* hoes printout *******
1020  rem ******************************
1030  open4,4:print#4,p$(1);p$(2);
1040  v$="-------------------
      ---":line$=v$+v$
1050  print#4,spc(4);line$
1060  fori=1to23:print#4,spc(4);"!";spc(
      70);"!":next
1070  print#4,spc(4);line$
1080  fori=1to28:print#4,spc(12);"!";spc
      (54);"!":next
1090  print#4,spc(12);left$(line$,56)
1100  close4
1110  return
2000  rem ******************************
2010  rem ***** directory inlezen *****
2020  rem ******************************
2030  gosub2600:print"[COM-3][3xCRSR-DOW
      N][2xSPACE]stop[SPACE]kant[SPACE]1
      [SPACE]van[SPACE]de[SPACE]disk[SPA
      CE]in[SPACE]de[SPACE]drive,"
2040  print"[CRSR-DOWN][7xSPACE]en[SPACE
      ]druk[SPACE]<return>[2xCRSR-DOWN][
      CTRL-4]"
2050  getq$:ifq$<>chr$(13)then2050
2060  gosub2240:print"[5xSPACE]":gosub2700
2070  forj=1tom:na$(j)=d$(j):next
2080  na$(0)=mid$(d$(0),1,17)+"id:"+mid$
      (d$(0),19,5)
2090  forj=m+1to99:na$(j)=left$(sp$,20):next
2095  forj=1to7:na$(99+j)=nn$(j):next
2100  na$(101)=left$(na$(101),15)+mid$(s
      tr$(c)+sp$,2,3)+right$(na$(101),2)
2110  na$(102)=left$(na$(102),15)+mid$(s
      tr$(i)+sp$,2,3)+right$(na$(102),2)
2120  na$(103)=left$(na$(103),15)+lo$+ri
      ght$(na$(103),2)
2130  gosub2600:print"[COM-3][3xCRSR-DOW
      N][2xSPACE]stop[SPACE]kant[SPACE]2
      [SPACE]van[SPACE]de[SPACE]disk[SPA
      CE]in[SPACE]de[SPACE]drive,"
2140  print"[CRSR-DOWN][7xSPACE]en[SPACE
      ]druk[SPACE]<return>[2xCRSR-DOWN][
      CTRL-4]"
2150  getq$:ifq$<>chr$(13)then2150
2160  gosub2240:print"[5xSPACE]":gosub2700
2170  forj=1tom:nb$(j)=d$(j):next
2180  nb$(0)=mid$(d$(0),1,17)+"id:"+mid$
      (d$(0),19,5)
2190  forj=m+1to99:nb$(j)=left$(sp$,20):next
2195  forj=1to7:nb$(99+j)=nn$(j):next
2200  nb$(101)=left$(nb$(101),15)+mid$(s
      tr$(c)+sp$,2,3)+right$(nb$(101),2)
2210  nb$(102)=left$(nb$(102),15)+mid$(s
      tr$(i)+sp$,2,3)+right$(nb$(102),2)
2220  nb$(103)=left$(nb$(103),15)+lo$+ri
      ght$(nb$(103),2)
2230  return
2240  rem ---------------------
2250  print"[CRSR-DOWN]"
2260  t=18:s=1:i=0
2270  open1,8,15,"i0"
2280  open2,8,2,"#"
2290  print#1,"u1";2;0;18;0
2300  input*2,255,a$
2310  lo$="on[SPACE]":ifmid$(a$,3,1)="a"
      thenlo$="off"
2320  d$(i)=mid$(a$,145,23)
2330  print#1,"u1";2;0;t;s
2340  input*2,255,a$
2350  t=asc(mid$(a$,1,1))
2360  s=asc(mid$(a$,2,1))
2370  ford=2to228step32
2380  i=i+1:ifi>144then2440
2390  printi"[CRSR-UP]":ifasc(mid$(a$,d+
      1,1))=0theni=i-1:goto2430
2400  d$(i)=mid$(a$,d+4,16)
2410  bl=asc(mid$(a$,d+28,1))*256+asc(mi
      d$(a$,d+29,1))
2420  d$(i)=d$(i)+right$("[2xSPACE]"+str
      $(bl),4)
2430  next:ift<>0then2330
2440  print#1,"m-r";chr$(250);chr$(2)
2450  get#1,c1$
2460  print#1,"m-r";chr$(252);chr$(2)
2470  get#1,c2$
2480  ifc1$=""thenc1$=chr$(0)
2490  ifc2$=""thenc2$=chr$(0)
2500  c=asc(c2$)*256+asc(c1$)
2510  close1:close2
2520  m=98:ifi<98thenm=i
2530  return
2600  rem
2610  print"[SHIFT-CLR][COM-6]"chr$(142)
      ;:printspc(tt)t1$:printspc(tt)t2$
2620  return
2700  rem ---------------------
2710  nn$=mid$(d$(0),1,16)
2720  ii$=mid$(d$(0),19,5)
2730  print"[CTRL-4][3xSPACE]naam:[SPACE
```

```
         ]"nn$
2740    print"[CRSR-DOWN][5xSPACE]id:[SPAC
         E]"ii$
2750    print"[
2760    getq$:ifq$<>"j"andq$<>"n"then2760
2770    ifq$="n"thenreturn
2800    poke19,1
2810    input"[6xCRSR-UP][9xCRSR-RIGHT]";nn$
2820    print:input"[CRSR-DOWN][9xCRSR-RIG
         HT]";ii$
2830    d$(0)=left$(nn$+sp$,18)+left$(ii$+
         sp$,5)
2840    return
3000    rem ******************************
3010    rem ***** directory printout *****
3020    rem ******************************
3030    open4,4:print#4
3040    print#4,p$(3);p$(4);p$(5);p$(6);
3050    print#4,spc(13)"O[25xCOM-Y]P[26xCO
         M Y]P"
3060    print#4,spc(13)"[COM-H]"na$(0)"[CO
         M N][SPACE]"nb$(0)"[COM-N]"

3070    print#4,spc(13)"L[25xCOM-P][SHIFT-
         '][26xCOM-P][SHIFT-']"
3080    print#4,p$(7);p$(8);p$(9);p$(10);p
         $(11);
3090    print#4,spc(68)"B"
3100    fori=0to52
3110    print#4,spc(24)na$(1+i)"[SPACE]+[S
         PACE]"na$(54+i)"[SPACE]B[SPACE]"nb
         $(1+i)"[SPACE]+[SPACE]"nb$(54+i)
3120    next
3130    print#4,spc(68)"B"
3140    print#4,p$(12);p$(13);p$(14);
3150    print#4:close4
3160    return
4000    rem ******************************
4010    rem ********* directory *********
4020    rem ******************************
4030    print"[SHIFT-CLR][COM-6]"chr$(142);
4040    printspc(14)"B[SPACE]directory[SPA
         CE]B"
4050    printspc(14)"JCCCCCCCCCCCK"
4060    print"[COM-3][3xCRSR-DOWN][2xSPACE
         ]stop[SPACE]de[SPACE]diskette[SPAC
         E]in[SPACE]de[SPACE]drive,"
4070    print"[CRSR-DOWN][6xSPACE]en[SPACE
         ]druk[SPACE]<return>[2xCRSR-DOWN][
         CTRL-4]"
4080    getq$:ifq$<>chr$(13)then4080
4090    gosub2240
4100    print"[SHIFT-CLR][CTRL-8]":j=0
4110    printd$(0):print"——————————
         ——"
4120    j=j+1:ifj>ithen4160
4125 print"[SPACE]"d$(j)
4130    if(j/20)<>int(j/20)then4150
4140    printspc(30)"[COM-3]<toets>":poke1
         98,0:wait198,1:printspc(30)"[CRSR-
         UP][7xSPACE][CRSR-UP][CTRL-8]"
4150    goto4120
4160    print"————————————-":pri
         nt"[2xSPACE]"c"blocks[SPACE]free.[
         13xCRSR-UP]"
4170    printspc(30)"[COM-3]<return>[CRSR-
         DOWN]":printspc(32)"voor[CRSR-DOWN
         ]":printspc(32)"menu."
4180    getq$:ifq$<>chr$(13)then4180
4190    return
5000    rem ******************************
5010    rem **** karakters definieren ****
5020    rem ******************************
5030    open4,4
5040    print#4,p$(15)
5050    print#4,p$(16)
5060    close4
5070    restore:be=91:en=126:gosub5150
5080    forw=1to11:readww:next
5090    be=160:en=191:gosub5150
5100    restore:forw=1to5*11:readww:next
5110    be=192:en=223:gosub5150
5120    be=224:en=254:gosub5150
5130    forw=1to11:readww:next
5140    return
5150    open4,4
5160    print#4,p$(17)chr$(be)chr$(en);
5170    forn=betoen
5180    print#4,chr$(139);
5190    form=1to11
5200    readmm
5210    print#4,chr$(mm);
5220    nextm:nextn
5230    print#4:close4
5240    return
5250    rem  ***** char. set one *****
5260    data0,0,0,254,130,130,130,130,0,0,0
5270    data0,18,18,126,146,146,130,130,13
         0,66,64
5280    data0,0,0,130,130,130,130,254,0,0,0
5290    data0,0,16,48,112,254,112,48,16,0,0
5300    data16,16,56,56,124,16,16,16,16,16,0
5310    data24,24,24,24,24,24,24,24,24,24,24
5320    data0,0,24,58,126,254,126,58,24,0,0
5330    data0,0,0,0,255,255,255,0,0,0,0
5340    data24,24,24,24,24,24,24,24,24,24,24
5350    data48,48,48,48,48,48,48,48,48,48,48
5360    data96,96,96,96,96,96,96,96,96,96,96
5370    data12,12,12,12,12,12,12,12,12,12,12
5380    data0,0,255,255,255,0,0,0,0,0,0
5390    data0,0,0,0,0,0,255,255,255,0,0
5400    data24,24,28,14,15,7,3,0,0,0,0
5410    data0,0,0,0,192,224,240,112,56,24,24
5420    data24,24,56,112,240,224,192,0,0,0,0
5430    data255,255,255,3,3,3,3,3,3,3,3
5440    data192,224,240,112,56,60,28,14,15,7,3
5450    data3,7,15,14,28,60,56,112,240,224,192
5460    data255,255,255,192,192,192,192,19
         2,192,192,192
5470    data192,192,192,192,192,192,192,19
         2,255,255,255
5480    data0,56,124,254,254,254,254,254,1
         24,56,0
5490    data6,6,6,6,6,6,6,6,6,6,6
5500    data0,96,240,248,124,62,124,248,24
         0,96,0
5510    data0,255,255,255,0,0,0,0,0,0,0
5520    data0,0,0,0,3,7,15,14,28,24,24
5530    data195,231,255,126,60,60,60,126,2
         55,231,195
5540    data0,56,124,254,198,198,198,254,1
         24,56,0
5550    data16,56,56,18,222,238,222,18,16,56,56
5560    data0,0,0,0,0,0,0,255,255,255,0
5570    data0,16,56,124,254,254,254,124,56,16,0
5580    data24,24,24,24,255,255,255,24,24,24,24
5590    data204,204,204,51,51,51,0,0,0,0,0
5600    data0,0,0,0,255,255,255,0,0,0,0
5610    data32,64,64,126,64,64,64,126,64,128,0
5620    data128,192,192,224,240,240,248,25
         2,252,254,255
5630    rem  ***** char. set two *****
```

print-out print-out print-out print-out print-out

```
5640  data0,0,0,0,0,0,0,0,0,0,0
5650  data255,255,255,255,255,255,0,0,0,0,0
5660  data15,15,15,15,15,15,15,15,15,15,15
5670  data128,128,128,128,128,128,128,12
      8,128,128,128
5680  data1,1,1,1,1,1,1,1,1,1,1
5690  data255,0,0,0,0,0,0,0,0,0,0
5700  data204,204,204,51,51,51,204,204,2
      04,51,51
5710  data0,0,0,0,0,0,0,0,0,0,255
5720  data12,12,12,3,3,3,12,12,12,3,3
5730  data255,254,252,252,248,240,240,22
      4,192,192,128
5740  data0,0,0,0,0,0,0,0,255,255,255
5750  data0,0,0,0,255,255,255,24,24,24,24
5760  data0,0,0,0,0,15,15,15,15,15,15
5770  data0,0,0,0,248,248,248,24,24,24,24
5780  data24,24,24,24,31,31,31,0,0,0,0
5790  data3,3,3,3,3,3,3,3,3,3,3
5800  data0,0,0,0,31,31,31,24,24,24,24
5810  data24,24,24,24,248,248,248,24,24,24,24
5820  data24,24,24,24,31,31,31,24,24,24,24
5830  data24,24,24,24,255,255,255,0,0,0,0
5840  data255,255,255,0,0,0,0,0,0,0,0
5850  data255,255,255,255,0,0,0,0,0,0,0
5860  data0,0,0,0,0,0,0,255,255,255,255
5870  data192,192,192,192,192,192,192,19
      2,192,192,192
5880  data224,224,224,224,224,224,224,22
      4,224,224,224
5890  data7,7,7,7,7,7,7,7,7,7,7
5900  data3,3,3,3,3,3,3,3,255,255,255
5910  data15,15,15,15,15,15,0,0,0,0,0
5920  data0,0,0,0,0,240,240,240,240,240,240
5930  data24,24,24,24,248,248,248,0,0,0,0
5940  data240,240,240,240,240,240,0,0,0,0,0
5950  data240,240,240,240,240,255,15,15,
      15,15,15
6000  rem ******************************
6010  rem *********** input* ***********
6020  rem ******************************
6030  rem ——————————————
6040  fori=828to922
6050  readx:pokei,x:s=s+x:next
6060  data169,71,160,3,141,8,3,140,9,3,9
      6,32,115,0,201,133,240,6,32,121,0,76
6070  data231,167,32,115,0,201,172,240,6
      ,32,191,171,76,174,167,32,155,183,32,30
6080  data225,32,253,174,32,158,183,138,
      72,32,253,174,32,139,176,133,73,132,74
6090  data32,163,182,104,32,117,180,160,
      2,185,97,0,145,73,136,16,248,200,32,18
6100  data225,145,98,200,196,97,208,246,
      32,204,255,76,174,167
6110  ifs<>11096thenprint"fout[SPACE]in[
      SPACE]data[SPACE]!!":end
6120  sys828:print"ok[SPACE]!"
6130  return
```

** Einde listing floppy-hoes

Checksum Floppy-hoes

REGEL 1	165	REGEL 310	146	REGEL 820	62	REGEL 2160	252	REGEL 2710	102
REGEL 2	100	REGEL 320	81	REGEL 830	222	REGEL 2170	153	REGEL 2720	99
REGEL 3	155	REGEL 330	230	REGEL 840	85	REGEL 2180	54	REGEL 2730	147
REGEL 4	254	REGEL 340	243	REGEL 850	32	REGEL 2190	32	REGEL 2740	107
REGEL 5	8	REGEL 350	86	REGEL 860	17	REGEL 2195	247	REGEL 2750	222
REGEL 6	205	REGEL 360	127	REGEL 870	118	REGEL 2200	143	REGEL 2760	210
REGEL 7	160	REGEL 370	24	REGEL 880	88	REGEL 2210	152	REGEL 2770	121
REGEL 8	118	REGEL 380	24	REGEL 890	24	REGEL 2220	179	REGEL 2800	94
REGEL 9	73	REGEL 390	100	REGEL 900	2	REGEL 2230	142	REGEL 2810	47
REGEL 10	111	REGEL 400	68	REGEL 910	48	REGEL 2240	213	REGEL 2820	163
REGEL 11	182	REGEL 410	100	REGEL 920	72	REGEL 2250	238	REGEL 2830	197
REGEL 12	132	REGEL 420	98	REGEL 930	39	REGEL 2260	68	REGEL 2840	142
REGEL 13	251	REGEL 430	100	REGEL 940	40	REGEL 2270	175	REGEL 3000	123
REGEL 14	158	REGEL 440	173	REGEL 950	142	REGEL 2280	38	REGEL 3010	109
REGEL 15	246	REGEL 450	182	REGEL 1000	81	REGEL 2290	166	REGEL 3020	123
REGEL 16	126	REGEL 460	176	REGEL 1010	143	REGEL 2300	188	REGEL 3030	57
REGEL 17	99	REGEL 470	229	REGEL 1020	81	REGEL 2310	193	REGEL 3040	202
REGEL 18	52	REGEL 480	153	REGEL 1030	200	REGEL 2320	139	REGEL 3050	83
REGEL 19	47	REGEL 490	32	REGEL 1040	154	REGEL 2330	180	REGEL 3060	104
REGEL 20	207	REGEL 500	176	REGEL 1050	130	REGEL 2340	188	REGEL 3070	140
REGEL 30	227	REGEL 510	151	REGEL 1060	88	REGEL 2350	87	REGEL 3080	99
REGEL 40	207	REGEL 520	107	REGEL 1070	130	REGEL 2360	87	REGEL 3090	59
REGEL 50	213	REGEL 530	164	REGEL 1080	142	REGEL 2370	247	REGEL 3100	183
REGEL 60	210	REGEL 540	32	REGEL 1090	97	REGEL 2380	232	REGEL 3110	35
REGEL 70	62	REGEL 550	200	REGEL 1100	212	REGEL 2390	240	REGEL 3120	130
REGEL 80	101	REGEL 560	136	REGEL 1110	142	REGEL 2400	21	REGEL 3130	59
REGEL 90	85	REGEL 570	72	REGEL 2000	81	REGEL 2410	36	REGEL 3140	36
REGEL 100	165	REGEL 580	75	REGEL 2010	253	REGEL 2420	107	REGEL 3150	218
REGEL 110	185	REGEL 590	77	REGEL 2020	81	REGEL 2430	158	REGEL 3160	142
REGEL 120	165	REGEL 600	136	REGEL 2030	57	REGEL 2440	116	REGEL 4000	81
REGEL 140	83	REGEL 610	36	REGEL 2040	210	REGEL 2450	185	REGEL 4010	56
REGEL 150	2	REGEL 700	2	REGEL 2050	158	REGEL 2460	118	REGEL 4020	81
REGEL 200	104	REGEL 710	215	REGEL 2060	252	REGEL 2470	186	REGEL 4030	243
REGEL 210	189	REGEL 720	152	REGEL 2070	152	REGEL 2480	82	REGEL 4040	74
REGEL 220	76	REGEL 730	85	REGEL 2080	53	REGEL 2490	84	REGEL 4050	7
REGEL 230	97	REGEL 740	17	REGEL 2090	31	REGEL 2500	71	REGEL 4060	152
REGEL 240	237	REGEL 750	17	REGEL 2095	246	REGEL 2510	221	REGEL 4070	210
REGEL 250	215	REGEL 760	116	REGEL 2100	140	REGEL 2520	145	REGEL 4080	163
REGEL 260	80	REGEL 770	227	REGEL 2110	149	REGEL 2530	142	REGEL 4090	85
REGEL 270	57	REGEL 780	78	REGEL 2120	176	REGEL 2600	213	REGEL 4100	116
REGEL 280	94	REGEL 790	142	REGEL 2130	58	REGEL 2610	218	REGEL 4110	164
REGEL 290	139	REGEL 800	2	REGEL 2140	210	REGEL 2620	142	REGEL 4120	156
REGEL 300	81	REGEL 810	93	REGEL 2150	159	REGEL 2700	213	REGEL 4125	224

vervolg Checksum Floppy-hoes

REGEL 4130	105	REGEL 5180	232	REGEL 5430	167	REGEL 5680	86	REGEL 5930	109
REGEL 4140	81	REGEL 5190	183	REGEL 5440	9	REGEL 5690	183	REGEL 5940	175
REGEL 4150	80	REGEL 5200	33	REGEL 5450	9	REGEL 5700	189	REGEL 5950	195
REGEL 4160	241	REGEL 5210	229	REGEL 5460	239	REGEL 5710	183	REGEL 6000	123
REGEL 4170	96	REGEL 5220	217	REGEL 5470	239	REGEL 5720	140	REGEL 6010	229
REGEL 4180	164	REGEL 5230	218	REGEL 5480	166	REGEL 5730	217	REGEL 6020	123
REGEL 4190	142	REGEL 5240	142	REGEL 5490	141	REGEL 5740	143	REGEL 6030	213
REGEL 5000	123	REGEL 5250	77	REGEL 5500	119	REGEL 5750	103	REGEL 6040	95
REGEL 5010	96	REGEL 5260	70	REGEL 5510	143	REGEL 5760	143	REGEL 6050	205
REGEL 5020	123	REGEL 5270	158	REGEL 5520	102	REGEL 5770	109	REGEL 6060	26
REGEL 5030	51	REGEL 5280	70	REGEL 5530	65	REGEL 5780	191	REGEL 6070	196
REGEL 5040	35	REGEL 5290	100	REGEL 5540	187	REGEL 5790	108	REGEL 6080	176
REGEL 5050	36	REGEL 5300	169	REGEL 5550	80	REGEL 5800	191	REGEL 6090	151
REGEL 5060	212	REGEL 5310	157	REGEL 5560	143	REGEL 5810	69	REGEL 6100	121
REGEL 5070	19	REGEL 5320	110	REGEL 5570	62	REGEL 5820	151	REGEL 6110	178
REGEL 5080	236	REGEL 5330	143	REGEL 5580	63	REGEL 5830	103	REGEL 6120	18
REGEL 5090	124	REGEL 5340	157	REGEL 5590	31	REGEL 5840	143	REGEL 6130	14r
REGEL 5100	147	REGEL 5350	223	REGEL 5600	143	REGEL 5850	251		
REGEL 5110	125	REGEL 5360	0	REGEL 5610	25	REGEL 5860	251		
REGEL 5120	125	REGEL 5370	124	REGEL 5620	217	REGEL 5870	239		
REGEL 5130	236	REGEL 5380	143	REGEL 5630	101	REGEL 5880	195		
REGEL 5140	142	REGEL 5390	143	REGEL 5640	75	REGEL 5890	152		
REGEL 5150	51	REGEL 5400	102	REGEL 5650	211	REGEL 5900	167		
REGEL 5160	170	REGEL 5410	144	REGEL 5660	157	REGEL 5910	143		
REGEL 5170	63	REGEL 5420	144	REGEL 5670	228	REGEL 5920	175		

Propschrift

Onder deze titel heeft Dim van den Heuvel veel meer verborgen dan je zou verwachten. Propschrift is een afkorting van Proportioneel schrift. Het programma voegt een extra commando aan Basic toe: POS. Met deze opdracht zet U teksten op het beeldscherm in een proportioneel schrift. Natuurlijk heeft dit voordelen:

- Er kunnen nu gemiddeld 45 tekens op het beeldscherm
- De letters hebben een natuurlijke stuktuur
- Voor weergave wordt de bitmap gebruikt zodat er eenvoudig plaatjes bij gezet kunnen worden.

Het programma kan bijzonder goed gebruikt worden voor bijvoorbeeld:

- handleidingen en/of uitleg bij een spel
- weergeven van tekst in een adventure

De machinetaal-data kan direkt worden weggeschreven van uit het basic-programma. Alle informatie om met het programma te kunnen werken staat in het programma zelf.

```
10    rem propschrift - commodore64
20    rem door : dim van den heuvel
30    rem        terheijden (nb)
100   poke53280,12:print"[HOME]";
110   gosub400
120   sys51513:rem karakterset positione
      ren
130   sys51419:rem runstop/restore bevei
      ligen
140   poke51101,28:rem code voor kleur
150   sys51090:rem lo-res ($cc00) vullen
       met kleurcode (28)
160   sys51064:rem hi-res ($e000) schoon
170   sys51162:rem het statement 'pos' a
      ctiveren
180   sys51127:rem bitmap aanzetten
190   print"[HOME]lllllllllllllllll":pr
      int"[CRSR-DOWN]ll[CRSR-DOWN][2xCRS
      R-LEFT]ll[CRSR-DOWN][2xCRSR-LEFT]l
      l[CRSR-DOWN][2xCRSR-LEFT]ll[CRSR-D
      OWN][2xCRSR-LEFT]ll[CRSR-DOWN][2xC
      RSR-LEFT]ll[2xCRSR-DOWN][2xCRSR-LE
      FT]";
195   print"llllllllllllllll":print"[CR
      SR-DOWN]ll[CRSR-DOWN][2xCRSR-LEFT]
      ll[CRSR-DOWN][2xCRSR-LEFT]ll[CRSR-
      DOWN][2xCRSR-LEFT]ll[CRSR-DOWN][2x
      CRSR-LEFT]ll[CRSR-DOWN][2xCRSR-LEF
      T]ll[CRSR-DOWN][2xCRSR-LEFT]ll[CRS
      R-DOWN][2xCRSR-LEFT]ll[CRSR-DOWN][
      2xCRSR-LEFT]ll"
197   print"[4xCRSR-DOWN]llllllllllllll
      lllllllllllllllll";:rem kleuren (
      l=zwart)
200   pos"Proportioneel[SPACE]schrift:",
      0,0
205   pos"*)[SPACE]Iedere[SPACE]letter[S
      PACE]heeft[SPACE]z'n[SPACE]eigen[S
      PACE]breedte.",0,2
210   pos"*)[SPACE]De[SPACE]natuurlijke[
      SPACE]letter-structuur[SPACE]bevor
      dert",0,3
215   pos"de[SPACE]leesbaarheid.",18,4
220   pos"*)[SPACE]Er[SPACE]kunnen[SPACE
      ]gemiddeld[SPACE]47[SPACE]karakter
      s[SPACE]op[SPACE]een[SPACE]regel."
      ,0,5
225   pos"*)[SPACE]Uitermate[SPACE]gesch
      ikt[SPACE]voor[SPACE]het[SPACE]wee
      rgeven[SPACE]van",0,6
230   pos"uitleg,[SPACE]handleidingen[SP
      ACE]en[SPACE]menu's.",18,7
235   pos"Technische[SPACE]gegevens:",0,
      9
240   pos"*)[SPACE]Het[SPACE]kleurgeheug
      en[SPACE]staat[SPACE]op[SPACE]loca
      tie[SPACE]$CC00.",0,11
245   pos"*)[SPACE]U[SPACE]kunt[SPACE]vo
      or-[SPACE]en[SPACE]achtergrondkleu
      ren[SPACE]instellen",0,12
250   pos"met:[SPACE]POKE51101,K:SYS5109
      0,[SPACE]waarbij",18,13
255   pos"K[SPACE]=[SPACE]16[SPACE]*[SPA
      CE]Voorgrondkleur[SPACE]+[SPACE]ac
      htergrondkleur.",18,14
260   pos"*)[SPACE]De[SPACE]bitmap[SPACE
```

```
      ]staat[SPACE]op[SPACE]locatie[SPAC
      E]$E000.",0,15
265   pos"*)[SPACE]U[SPACE]zet[SPACE]een
      [SPACE]tekst[SPACE]op[SPACE]het[SP
      ACE]scherm[SPACE]met:",0,16
270   a$="POS"+chr$(34)+"tekst"+chr$(34)
      +",x,y[SPACE]waarbij[SPACE]x[SPACE
      ]varieert[SPACE]van[SPACE]0-319"
275   posa$,18,17
280   pos"en[SPACE]y[SPACE]van[SPACE]0-2
      4[SPACE](geeft[SPACE]de[SPACE]REGE
      L[SPACE]aan!)",18,18
285   pos"*)[SPACE]Voor[SPACE]efficient[
      SPACE]gebruik[SPACE]kunt[SPACE]u[S
      PACE]het[SPACE]beste[SPACE]de",0,1
      9
290   pos"machinetaal[SPACE]direct[SPACE
      ]wegschrijven,[SPACE]en[SPACE]vanu
      it",18,20
295   pos"uw[SPACE]eigen[SPACE]programma
      [SPACE]inladen.[SPACE]Het[SPACE]pr
      ogramma",18,21
300   pos"beslaat[SPACE]de[SPACE]locatie
      s[SPACE]$C700[SPACE]tot[SPACE]$ca7
      5.",18,22
305   pos"Druk[SPACE]op[SPACE]RETURN[SPA
      CE]voor[SPACE]het[SPACE]volgende[S
      PACE]scherm.",0,24
310   getr$:ifr$<>chr$(13)then310
315   sys51064:sys51090
320   print"[HOME]ll[5xCRSR-DOWN][2xCRSR
      -LEFT]ll[3xCRSR-DOWN][2xCRSR-LEFT]
      ll[3xCRSR-DOWN][2xCRSR-LEFT]ll"
330   pos"*)[SPACE]U[SPACE]kunt[SPACE]de
      [SPACE]machinetaal-data[SPACE]wegs
      chrijven[SPACE]door",0,0
335   pos"op[SPACE]'S'[SPACE]te[SPACE]dr
      ukken.[SPACE]U[SPACE]kunt[SPACE]de
      [SPACE]routine[SPACE]dan[SPACE]van
      uit",18,1
340   pos"uw[SPACE]eigen[SPACE]programma
      's[SPACE]inlezen[SPACE]op[SPACE]de
      ze[SPACE]manier:",18,2
345   pos"Laat[SPACE]uw[SPACE]programma[
      SPACE]beginnen[SPACE]met:",18,3
350   a$="10[SPACE]IFA=1THEN30[3xSPACE]2
      0[SPACE]A=1:LOAD"+chr$(34)+"PRM"+c
      hr$(34)+",8,1"
355   a$=a$+"[2xSPACE]30[SPACE]...":posa
      $,18,4
360   pos"*)[SPACE]De[SPACE]benodigde[SP
      ACE]SYS-opdrachten[SPACE]staan[SPA
      CE]in[SPACE]de",0,5
365   pos"eerste[SPACE]regels[SPACE]van[
      SPACE]het[SPACE]programma[SPACE]ve
      rmeld,",18,6
370   pos"behalve[SPACE]deze:[SPACE]SYS5
      1808[SPACE]=[SPACE]Uitzetten[SPACE
      ]bitmap.",18,7
372   pos"*)[SPACE]Alle[SPACE]tekst[SPAC
      E]staat[SPACE]in[SPACE]bitmap[SPAC
      E]en[SPACE]is[SPACE]dus[SPACE]makk
      elijk",0,8
374   pos"te[SPACE]combineren[SPACE]met[
      SPACE]Doodle-tekeningen[SPACE]en[S
      PACE]uit",18,9
376   pos"printen[SPACE]met[SPACE]een[SP
      ACE]grafisch[SPACE]programma.",18,
      10
378   pos"*)[SPACE]Type[SPACE]nu[SPACE]'
      S'[SPACE]voor[SPACE]SAVE[SPACE](de
      [SPACE]computer[SPACE]wordt",0,11
379   pos"na[SPACE]het[SPACE]saven[SPACE
      ]gereset!)[SPACE]of[SPACE]'E'[SPAC
      E]voor[SPACE]einde.",18,12
380   getr$:ifr$="s"then386
382   ifr$=""then380
384   sys51808:poke53280,14:end
385   getr$:ifr$=""then385
386   poke43,0:poke44,199:poke45,144:pok
      e46,202:save"prm",8,1:sys64738
390   sys51808:poke53280,14:end
400   rem data-inlees-routine
410   fort=0to751:reada:c=c+a:poke50944+
      t,a:next
420   ifc<>96120thenprint"fout[SPACE]in[
      SPACE]gedeelte[SPACE]a":end
430   c=0:fort=0to95:reada:c=c+a:poke517
      12+t,a:next
440   ifc<>10642thenprint"fout[SPACE]in[
      SPACE]gedeelte[SPACE]b":end
450   fort=0to20:reada:poke51808+t,a:nex
      t
460   return
499   rem data gedeelte a:
500   data32,158,173,32,163,182,168,140
502   data63,3,136,177,34,153,64,3,136
504   data16,248,32,253,174,32,235,183
506   data169,0,160,224,133,165,132,166
508   data165,20,164,21,133,250,132,251
510   data224,0,240,16,24,165,165,105
512   data64,133,165,165,166,105,1,133
514   data166,202,208,240,165,20,133,250
516   data165,21,133,251,169,0,133,2
518   data165,250,133,20,165,251,133,21
520   data164,2,185,64,3,32,0,201
522   data32,0,200,24,165,250,109,167
524   data2,133,250,165,251,105,0,133
526   data251,230,2,164,2,204,63,3
528   data208,214,96,208,214,96,234,162
530   data32,160,224,169,0,133,250,132
532   data251,160,0,169,0,145,250,200
534   data208,249,230,251,202,208,242,96
536   data234,169,0,160,204,133,250,132
538   data251,160,0,169,28,145,250,165
540   data250,201,231,208,7,165,251,201
542   data207,208,1,96,230,250,208,2
544   data230,251,76,156,199,234,169,9
546   data141,17,208,234,234,234,234,234
548   data234,234,234,169,148,141,0,221
550   data169,56,141,24,208,169,204,141
552   data136,2,169,59,141,17,208,96
554   data234,169,229,141,8,3,169,199
556   data141,9,3,96,32,115,0,201
558   data185,240,3,76,248,199,32,115
560   data0,32,0,199,76,174,167,32
562   data121,0,76,231,167,234,234,168
564   data133,180,32,241,200,169,0,133
566   data181,24,6,180,38,181,6,180
568   data38,181,6,180,38,181,24,165
570   data181,105,208,133,181,185,0,212
572   data141,167,2,169,0,141,168,2
574   data165,21,74,165,20,106,141,169
576   data2,110,168,2,78,169,2,110
578   data168,2,78,169,2,110,168,2
580   data78,168,2,78,168,2,78,168
582   data2,78,168,2,78,168,2,173
584   data169,2,133,20,169,0,133,21
586   data6,20,38,21,6,20,38,21
588   data6,20,38,21,24,165,165,101
590   data20,133,20,165,166,101,21,133
592   data21,160,0,173,168,2,208,7
```

```
594    data177,180,145,20,76,209,200,169
596    data0,141,171,2,177,180,141,170
598    data2,174,168,2,78,170,2,110
600    data171,2,202,208,247,169,0,141
602    data172,2,169,255,141,173,2,174
604    data168,2,56,110,172,2,110,173
606    data2,202,208,246,177,20,45,172
608    data2,13,170,2,145,20,200,200
610    data200,200,200,200,200,200,177,20
612    data45,173,2,13,171,2,145,20
614    data136,136,136,136,136,136,136,13
       6
616    data200,192,8,208,158,32,45,201
618    data96,234,120,169,232,141,24,3
620    data169,200,141,25,3,88,96,169
622    data4,141,136,2,76,71,254,234
624    data120,169,0,141,14,220,169,48
626    data133,1,96,234,234,234,234,16
628    data18,56,233,128,201,32,16,2
630    data169,128,96,201,64,16,249,24
632    data105,64,96,201,32,16,3,169
634    data128,96,201,64,16,1,96,201
636    data96,16,4,56,233,64,96,56
638    data233,32,96,234,169,55,133,1
640    data169,1,141,14,220,88,96,234
642    data120,169,0,141,14,220,169,51
644    data133,1,162,4,169,0,160,216
646    data133,250,132,251,169,0,160,204
648    data133,252,132,253,160,0,177,250
650    data145,252,200,208,249,230,251,23
       0
652    data253,202,208,240,169,48,133,1
654    data32,154,201,162,4,169,0,160
656    data204,133,250,132,251,169,0,160
658    data208,133,252,132,253,160,0,177
660    data250,145,252,200,208,249,230,251
662    data230,253,202,208,240,169,55,133
664    data1,169,1,141,14,220,88,96
666    data234,162,96,169,8,157,0,212
668    data232,224,128,208,246,169,0,160
670    data204,133,250,132,251,162,0,160
672    data0,189,0,202,24,41,240,106
674    data106,106,106,157,0,212,189,0
676    data202,41,15,133,2,240,19,160
678    data0,177,250,24,42,145,250,200
680    data192,8,208,245,198,2,165,2
682    data208,237,24,165,250,105,8,133
684    data250,165,251,105,0,133,251,232
686    data224,96,208,195,96,234,234
687    rem data gedeelte b:
688    data113,113,113,113,113,113,98,113
690    data113,82,98,113,82,129,113,113
692    data113,113,113,113,113,113,113,129
694    data113,113,113,97,128,97,113,113
696    data80,82,113,128,113,113,113,98,82
698    data82,128,113,82,113,82,129,113
700    data113,113,113,113,113,113,113,113
702    data113,82,81,113,113,113,113,128
704    data113,113,113,113,113,113,113,113
706    data82,113,113,113,129,113,113,113
708    data113,113,113,113,113,113,129,113
710    data113,113,128,128,128,128,128
711    rem gedeelte c
712    data169,27,141,17,208,169,21,141,24
713    data208,169,151,141,0,221,169,4,141
714    data136,2,96
```

** EINDE LISTING propschrift

Checksum Propschrift

REGEL 10	31	REGEL 355	164	REGEL 544	2	REGEL 634	88
REGEL 20	92	REGEL 360	194	REGEL 546	76	REGEL 636	8
REGEL 30	82	REGEL 365	37	REGEL 548	26	REGEL 638	142
REGEL 100	141	REGEL 370	71	REGEL 550	36	REGEL 640	143
REGEL 110	33	REGEL 372	184	REGEL 552	152	REGEL 642	175
REGEL 120	70	REGEL 374	41	REGEL 554	209	REGEL 644	76
REGEL 130	211	REGEL 376	24	REGEL 556	231	REGEL 646	16
REGEL 140	210	REGEL 378	229	REGEL 558	200	REGEL 648	20
REGEL 150	136	REGEL 379	232	REGEL 560	43	REGEL 650	119
REGEL 160	35	REGEL 380	227	REGEL 562	237	REGEL 652	233
REGEL 170	221	REGEL 382	57	REGEL 564	220	REGEL 654	125
REGEL 180	206	REGEL 384	194	REGEL 566	90	REGEL 656	16
REGEL 190	14	REGEL 385	143	REGEL 568	146	REGEL 658	23
REGEL 195	193	REGEL 386	2	REGEL 570	21	REGEL 660	121
REGEL 197	121	REGEL 390	194	REGEL 572	84	REGEL 662	75
REGEL 200	116	REGEL 400	233	REGEL 574	236	REGEL 664	39
REGEL 205	212	REGEL 410	85	REGEL 576	31	REGEL 666	146
REGEL 210	237	REGEL 420	199	REGEL 578	44	REGEL 668	30
REGEL 215	2	REGEL 430	127	REGEL 580	69	REGEL 670	9
REGEL 220	153	REGEL 440	195	REGEL 582	4	REGEL 672	69
REGEL 225	68	REGEL 450	189	REGEL 584	76	REGEL 674	176
REGEL 230	5	REGEL 460	142	REGEL 586	131	REGEL 676	118
REGEL 235	170	REGEL 499	99	REGEL 588	77	REGEL 678	172
REGEL 240	187	REGEL 500	35	REGEL 590	215	REGEL 680	98
REGEL 245	77	REGEL 502	239	REGEL 592	30	REGEL 682	235
REGEL 250	68	REGEL 504	242	REGEL 594	37	REGEL 684	14
REGEL 255	140	REGEL 506	28	REGEL 596	174	REGEL 686	108
REGEL 260	200	REGEL 508	15	REGEL 598	33	REGEL 687	100
REGEL 265	204	REGEL 510	176	REGEL 600	179	REGEL 688	59
REGEL 270	40	REGEL 512	235	REGEL 602	190	REGEL 690	236
REGEL 275	71	REGEL 514	68	REGEL 604	126	REGEL 692	102
REGEL 280	224	REGEL 516	126	REGEL 606	183	REGEL 694	27
REGEL 285	18	REGEL 518	17	REGEL 608	55	REGEL 696	127
REGEL 290	23	REGEL 520	231	REGEL 610	36	REGEL 698	235
REGEL 295	11	REGEL 522	175	REGEL 612	24	REGEL 700	95
REGEL 300	178	REGEL 524	168	REGEL 614	135	REGEL 702	14
REGEL 305	139	REGEL 526	21	REGEL 616	182	REGEL 704	95
REGEL 310	109	REGEL 528	41	REGEL 618	184	REGEL 706	59
REGEL 315	117	REGEL 530	223	REGEL 620	152	REGEL 708	102
REGEL 320	185	REGEL 532	169	REGEL 622	86	REGEL 710	188
REGEL 330	201	REGEL 534	82	REGEL 624	181	REGEL 711	17
REGEL 335	250	REGEL 536	17	REGEL 626	185	REGEL 712	131
REGEL 340	246	REGEL 538	237	REGEL 628	79	REGEL 713	121
REGEL 345	75	REGEL 540	15	REGEL 630	202	REGEL 714	2
REGEL 350	85	REGEL 542	130	REGEL 632	88		

Reken-Motief

Ook deze inzender heeft reeds de nodige programma's ontworpen. De Heer Y.S. Kasmin uit Amsterdam. Het is een rekenprogramma waarbij een locomotief een hoofdrol speelt. Het is een rekenprogramma waarbij de moeilijkheidsgraad kan worden ingesteld. Door de grafische beelden bij dit geheel gaat het niet zo snel vervelen.

```
1    rem ********************************
2    rem **         reken-motief       **
3    rem **            door            **
4    rem **  door:  y.s.kasmin         **
5    rem **      ****************      **
6    rem ********** a'dam-oost ********
10   gosub850:rem ****sprite-inlezen
20   poke53280,5:poke53281,7:print"[SHI
     FT CLR]"spc(7)"[3xCRSR-DOWN][CTRL
     9][COM-1][SPACE]r[SPACE]e[SPACE]k[
     SPACE]e[SPACE]n[SPACE]-[SPACE]m[SP
     ACE]o[SPACE]t[SPACE]i[SPACE]e[SPAC
     E]f[SPACE][CTRL-0]"
30   printchr$(142):printspc(16)"[CRSR-
     DOWN][COM-2]door[CRSR-DOWN]":print
     spc(13)"[CTRL-7]y.s.kasmin[2xCRSR-
     DOWN]"
40   printspc(9)"[2xSPACE][COM-4]a'dam-
     oost[SPACE]1988":printspc(7)"[3xCR
     SR-DOWN][CTRL-5]>>[SPACE]toets[SPA
     CE]willekeurig[SPACE]<<"
50   geta$:ifa$=""then50
60   si=54272:en=5:open1,0:print"[SHIFT
      CLR]"chr$(8)chr$(14)
70   z$="[CTRL-9][CTRL-3][2xCOM-U][CRSR
     -DOWN][3xCRSR-LEFT][4xSPACE][CRSR-
     DOWN][5xCRSR-LEFT][6xSPACE][CRSR-D
     OWN][7xCRSR-LEFT][2xSPACE][CTRL-6]
     [2xSPACE][CTRL-3][SPACE][CTRL-2][S
     PACE][CTRL-3][2xSPACE][CRSR-DOWN][
     8xCRSR-LEFT][2xSPACE][CTRL-6][2xSP
     ACE][CTRL-3][4xSPACE][CRSR-DOWN][8
     xCRSR-LEFT]"
80   z1$="[2xSPACE][CTRL-6][2xSPACE][CT
     RL 3][4xSPACE][CRSR-DOWN][9xCRSR-L
     EFT][COM-8][CTRL-9][COM-1][COM-Q][
     COM-W][COM-Q][COM-W][COM-Q][COM-W]
     [COM-Q][COM-W][COM-Q][COM-W]":x=24
90   fori=sitosi+24:pokesi,0:next:pokes
     i+24,15:pokesi,86:pokesi+6,240
100  rem ********************************
110  rem *************menu**************
120  rem ********************************
130  print"[SHIFT-CLR]":poke53280,4:pok
     e53281,7:poke646,6:printspc(10)"[C
     TRL 9][CTRL-1][SPACE]S[SPACE]P[SPA
     CE]E[SPACE]L-M[SPACE]E[SPACE]N[SPA
     CE]U[SPACE][CTRL-0][2xCRSR-DOWN]"
140  printspc(8)"[CTRL-9][CTRL-6][SPACE
     ]1[SPACE][CTRL-0][CTRL-7][SPACE]-[
     SPACE]Vermenigvuldigen[CRSR-DOWN]"
150  printspc(8)"[CTRL-9][CTRL-6][SPACE
     ]2[SPACE][CTRL-0][COM-2][SPACE]-[S
     PACE]Delen":print:printspc(8)"[CTR
     L 9][CTRL-6][SPACE]3[SPACE][CTRL-0
     ][CTRL-1][SPACE]-[SPACE]Optellen[C
     RSR-DOWN]"
160  printspc(8)"[CTRL-9][CTRL-6][SPACE
     ]4[SPACE][CTRL-0][COM-3][SPACE]-[S
     PACE]Aftrekken":print:printtab(8)"
     [CTRL-9][CTRL-6][SPACE]5[SPACE][CT
     RL 0][CTRL-5][SPACE]-[SPACE]Uitleg
     [SPACE][CRSR-DOWN]"
170  printspc(8)"[CTRL-9][CTRL-6][SPACE
     ]6[SPACE][CTRL-0][SPACE]-[SPACE]St
     oppen[2xCRSR-DOWN]"
180  printspc(4)"[CTRL-9][COM-7][SPACE]
     Tik[SPACE]het[SPACE]nummer[SPACE]v
     an[SPACE]uw[SPACE]keuze...[SPACE]"
190  getkz$:ifkz$=""then190
200  kz=val(kz$):ifkz<1orkz>6then190
210  ifkz=6thenprint"[SHIFT-CLR]":end
220  ifkz=5thengosub740:goto130
230  gosub270:gosub370:gosub450:goto470
240  rem *******************************
     *
250  rem ***invoer maximale getallen***
     *
260  rem ***    om mee te rekenen     ***
     *
270  poke53280,0:poke53281,0:print"[SHI
     FT CLR][3xCRSR-DOWN]"tab(7)"[CTRL
     2]Geef[SPACE]het[SPACE]maximuum...
     ";:input#1,s
280  rem *******************************
290  rem *******random getallen********
300  print"[SHIFT-CLR]"
310  a=int(rnd(0)*s)+1:b=int(rnd(0)*s)+
     1:t$="=[SPACE]"
320  ifkz=1thenga=a*b:rt$="X"
330  ifkz=2thenga=a/b:rt$=":"
340  ifkz=3thenga=a+b:rt$="+"
350  ifkz=4thenga=a-b:rt$="-"
360  return
370  rem *******************************
380  rem *********scherm-opmaak********
390  print"[CTRL-8][3xCRSR-DOWN][3xCRSR
     -RIGHT]SCHERM[SHIFT-SPACE]WORDT[SH
     IFT SPACE]EVEN[SHIFT-SPACE]UITGESC
     HAKELD":fori=0to1500:next:poke5326
     5,0
400  print"[SHIFT-CLR]":fori=1024to1024
     +(40*5)-1:cr=cr+1:poke55295+cr,8:p
     okei,160:next
410  poke211,5:poke214,2:sys58640
420  print"[CTRL-9][CTRL-8][SPACE]SCORE
     :"sc;tab(24)"[CTRL-9][CTRL-8][SPAC
     E]ENERGIE:"en
430  ifen=0thengoto680
440  return
450  fori=1664to1664+39:pokei,113+128:c
     r=cr+1:poke55736+cr,int(rnd(0)*15)
     +1:next
460  poke211,34:poke214,9:sys58640:prin
     tz$+z1$:poke53269,1:poke53265,27:r
     eturn
470  rem ********************************
480  rem ********begin spel*************
490  poke211,3:poke214,18:sys58640:prin
     t "[CTRL-6]Hoeveel[SPACE]is...:"a;
     rt$;b;t$;
500  input#1,an
510  ifan<>gathengosub530:gosub310:goto470
520  ifan=gathengosub580:gosub310:goto470
530  rem *******fout geantwoord********
540  poke211,10:poke214,20:sys58640:pri
     nt"[CTRL-9][CTRL-8][SPACE]fout!!!!
     [SPACE][CTRL-0]":en=en-1:ps=1500
550  poke211,3:poke214,22:sys58640:prin
     t"[COM-2]Het[SPACE]goede[SPACE]ant
     woord[SPACE]was..:[CTRL-9][CTRL-6]
```

```
     "ga:gosub410
560  fori=0tops:next:fori=17to24:poke78
     1,i:sys59903:next:return
570  rem ******antwoord is goed********
580  x=x+5:x1=x-(xand256):v=53248:si=54
     272
590  if x>255thenpokev+16,1
600  ifx=279thengoto640
610  pokev,x1:pokesi+4,17:fori=198to0st
     ep-3:pokesi+1,i:next:pokesi+4,0:sc
     =sc+5
620  ps=0:gosub560:gosub410:return
630  rem ********doel bereikt********
640  poke211,1:poke214,6:sys58640:print
     "[CTRL-9][COM-5][SPACE]GEFELICITEE
     RD[SHIFT-SPACE]HET[SHIFT-SPACE]STA
     TION[SPACE]IS[SPACE]BEREIKT[SPACE]
     [CTRL-0]
650  printspc(9)"[CRSR-DOWN][CTRL-2]Toe
     ts[SPACE]willekeurig"
660  getwk$:ifwk$=""then660
670  sc=0:en=5:pokev+16,0:pokev+21,0:pr
     int"[SHIFT-CLR]":close1:clr:goto 6
     0
680  rem ******energie is op*******
690  poke211,5:poke214,6:sys58640
700  print "[CTRL-9][CTRL-6][SPACE]H[SP
     ACE]E[SPACE]L[SPACE]A[SPACE]A[SPAC
     E]S[2xSPACE]JE[SPACE]ENERGIE[SPACE
     ][CRSR-DOWN]":printtab(13)"[CTRL-9
     ][CTRL-5][SPACE]IS[SPACE]OP[SPACE]
     [CTRL-0][CRSR-DOWN]"
710  printtab(8)"[CTRL-7]Toets[SPACE]wi
     llekeurig"
720  getwk$:ifwk$=""then720
730  sc=0:en=5:pokev+16,0:pokev+21,0:pr
     int"[SHIFT-CLR]":close1:clr:goto60
740  fori=17to0step-1:poke781,i:sys5990
     3:fori1=0to50:nexti1,i:poke53280,0
750  poke53281,0:print"[HOME]"spc(12)"[
     CTRL-9][CTRL-8][SPACE]U[SPACE]I[SH
     IFT SPACE]T[SHIFT-SPACE]L[SHIFT-SP
     ACE]E[SHIFT-SPACE]G[SPACE][CTRL-0]
     [2xCRSR-DOWN]":poke646,5
760  printspc(2)"1.[SPACE]Kies[SPACE]ui
     t[SPACE]het[SPACE]MENU.[CRSR-DOWN]
     ":printspc(2)"2.[SPACE]";
770  print"Probeer[SPACE]bij[SPACE]elke
     [SPACE]som,[SPACE]een[SPACE]goed[S
     PACE]an[SPACE]-[5xSPACE]twoord[SPA
     CE]te[SPACE]geven[SPACE]zodat[SPAC
     E]";
780  print"de[SPACE][CTRL-3]LOCOMOTIEF[
     CTRL-6][5xSPACE]het[SPACE]station[
     SPACE]kan[SPACE]bereiken.[CRSR-DOW
     N]"
790  printspc(2)"3.[SPACE]Een[SPACE]goe
     d[SPACE]antwoord[SPACE]wordt[SPACE
     ]beloond[SPACE]met[5xSPACE]5[SPACE
     ]punten.[CRSR-DOWN]"
800  printspc(2)"4.[SPACE]Bij[SPACE]een
     [SPACE]foute[SPACE]antwoord[SPACE]
     verlies[SPACE]je[7xSPACE]een[SPACE
     ]energie.[2xCRSR-DOWN]"
810  printspc(5)"[CTRL-2]Druk[SPACE]na[
     SPACE]elk[SPACE]gegeven[SPACE]antw
     oord[SPACE]op...[15xSPACE][2xCRSR-
     DOWN][CTRL-9][CTRL-8][SPACE]RETURN
     [SHIFT-SPACE]"
820  printspc(7)"[2xCRSR-DOWN]Toets[SPA
     CE]willekeurig......"
830  getwk$:ifwk$=""then830
840  return
850  rem ******************************
860  rem *******set-up sprite*********
870  v=53248:fori=0to62:readd:poke832+i
     ,d:next:poke2040,13:pokev+28,1
880  pokev+37,12:pokev+38,6:pokev+39,7:
     pokev,24:pokev+1,136:pokev+29,1
890  pokev+23,1:return
900  data 2,0,0,0,128,0,0,34,0
910  data 0,128,0,0,8,128,0,0,0
920  data 0,2,0,0,0,64,0,0,64
930  data 0,21,64,0,31,80,84,31,212
940  data 100,23,244,101,85,245,106,169
     ,85
950  data 106,170,164,106,170,164,106,1
     70,165
960  data 85,85,85,7,65,208,5,65,80
```

** EINDE LISTING reken-motief

Checksum Reken-motief

REGEL 1	249	REGEL 500	112
REGEL 2	157	REGEL 510	139
REGEL 3	107	REGEL 520	222
REGEL 4	112	REGEL 530	61
REGEL 5	215	REGEL 540	235
REGEL 6	173	REGEL 550	213
REGEL 10	180	REGEL 560	35
REGEL 20	227	REGEL 570	4
REGEL 30	214	REGEL 580	130
REGEL 40	181	REGEL 590	50
REGEL 50	50	REGEL 600	1
REGEL 60	227	REGEL 610	75
REGEL 70	255	REGEL 620	11
REGEL 80	35	REGEL 630	131
REGEL 90	18	REGEL 640	243
REGEL 100	165	REGEL 650	78
REGEL 110	50	REGEL 660	43
REGEL 120	165	REGEL 670	4
REGEL 130	224	REGEL 680	235
REGEL 140	111	REGEL 690	53
REGEL 150	25	REGEL 700	173
REGEL 160	193	REGEL 710	83
REGEL 170	212	REGEL 720	40
REGEL 180	72	REGEL 730	4
REGEL 190	47	REGEL 740	255
REGEL 200	1	REGEL 750	22
REGEL 210	233	REGEL 760	227
REGEL 220	61	REGEL 770	218
REGEL 230	69	REGEL 780	242
REGEL 240	165	REGEL 790	235
REGEL 250	34	REGEL 800	46
REGEL 260	201	REGEL 810	241
REGEL 270	108	REGEL 820	108
REGEL 280	123	REGEL 830	42
REGEL 290	18	REGEL 840	142
REGEL 300	112	REGEL 850	123
REGEL 310	62	REGEL 860	238
REGEL 320	245	REGEL 870	81
REGEL 330	89	REGEL 880	232
REGEL 340	72	REGEL 890	33
REGEL 350	76	REGEL 900	55
REGEL 360	142	REGEL 910	113
REGEL 370	123	REGEL 920	9
REGEL 380	1	REGEL 930	65
REGEL 390	96	REGEL 940	177
REGEL 400	124	REGEL 950	66
REGEL 410	49	REGEL 960	11
REGEL 420	6		
REGEL 430	206		
REGEL 440	142		
REGEL 450	26		
REGEL 460	126		
REGEL 470	165		
REGEL 480	154		
REGEL 490	52		

Tone Dail

Dit programma van Bart Koop en Ronalt Aalmoes uit Schagen genereert de telefoon beltonen. Wanneer men nog een puls telefoon heeft, en men houd de hoorn bij de luidspreker, dan kan men via de toetsenbord van de commodore 64 bellen. Het programma kan ook uit geos opgeroepen worden. Het test zelf of dit programma aanwezig is.

```
1     rem - tone dial emulator  v2.5  -
2     rem -                           -
3     rem - door: bart koop, schagen  -
4     rem - (c) 1988  bakosoft        -
5     :
6     poke 53280,0:poke 53281,0:poke 646
      ,13:print chr$(147)"even[SPACE]ged
      uld[SPACE]aub[SPACE].[SPACE].[SPACE]."
7     :
8     rem - run/stop uit -
9     rem - nmi flag normaal -
10    :
11    poke 788,52:poke 792,71:poke 793,254
12    :
13    rem - reset & init. geluid -
14    :
15    s=54272:for k=0to24:poke s+k,0:next k
16    poke s+24,15:poke s+5,17:poke s+6,
      241:poke s+4,17
17    poke s+12,17:poke s+13,241:poke s+11,17
18    :
19    rem - data voor toon-freqs -
20    rem - & toetsen            -
21    :
22    data 35,60,82,85,160
23    data 56,44,176,77,125
24    data 59,44,176,85,160
25    data 8,44,176,94,169
26    data 11,49,93,77,125
27    data 16,49,93,85,160
28    data 19,49,93,94,169
29    data 24,54,158,77,125
30    data 27,54,158,85,160
31    data 32,54,158,94,169
32    data 49,60,82,77,125
33    data 1,60,82,94,169
34    data 4,44,176,104,167
35    data 5,49,93,104,167
36    data 6,54,158,104,167
37    data 3,60,82,104,167
38    :
39    rem - dimensioneer toets/toon -
40    rem - variabelen              -
41    :
42    dim tn(255,2,2)
43    :
44    rem - lees toets/toon variabelen -
45    :
46    for k=22to37
47    read t,h1,l1,h2,l2
48    tt=tt+t+h1+l1+h2+l2
49    tn (t,1,1)=h1:tn (t,1,2)=l1:tn (t,
      2,1)=h2:tn (t,2,2)=l2
50    next k
51    :
52    rem - mc-data (desktop) -
53    :
54    data32,113,194,169,8,32,176,194,32
      ,161,194,32,44,194
55    :
56    rem - lees mc-data -
57    :
58    fork=16384to16397
59    readl:pokek,l:tt=tt+l
60    nextk
61    :
62    rem - fout in dataregels? -
63    :
64    if tt<>8718 then print chr$(17);"?
      data[SPACE]error";chr$(145):poke 7
      88,49:end
65    :
66    rem - print tekst -
67    :
68    print chr$(147);chr$(17);chr$(29);
      chr$(29);"tone[SPACE]dial[SPACE]em
      ulator[SPACE]v2.5"
69    print chr$(17);chr$(29);chr$(29);"
      (c)[SPACE]1988[SPACE]bakosoft"
70    print chr$(17);chr$(17);chr$(29);c
      hr$(29);"#[SPACE]=[SPACE][return]"
71    print chr$(17);chr$(17);chr$(29);c
      hr$(29);"naar[SPACE]een[SPACE]idee
      [SPACE]van[SPACE]bart[SPACE]koop"
72    print chr$(17);chr$(29);chr$(29);"
      c-64[SPACE]versie[SPACE]:bart[SPAC
      E]koop"
73    print chr$(17);chr$(29);chr$(29);"
      amiga[SPACE]versie:roalt[SPACE]aal
      moes"
74    print chr$(17);chr$(17);chr$(29);c
      hr$(29);"runstop/restore[SPACE]voo
      r[SPACE]einde"
75    :
76    rem - geos aanwezig? -
77    :
78    if peek(49152)<>76 or peek(49153)<
      >16 goto83
79    print chr$(17);chr$(29);chr$(29);"
      [c=][SPACE]g[SPACE]=[SPACE]terug[S
      PACE]naar[SPACE]geos"
80    :
81    rem - scan toets/genereer toon -
82    :
83    t=peek(197):if t=64 goto 83
84    :
85    rem - terug naar geos? -
86    :
87    if peek(653)=2 and t=26 goto 110
88    :
89    rem - t=64 ==> geen toets -
90    :
91    poke s,tn (t,1,2):poke s+1,tn (t,1,
      1)
92    poke s+7,tn (t,2,2):poke s+8,tn (t,2,1)
93    :
94    rem - minimum duur toon -
95    :
96    for q=0to10:next q
97    :
98    rem - wacht op loslaten toets -
99    :
100   wait 197,64
101   :
102   rem - stop toon -
103   :
104   poke s,0:poke s+1,0:poke s+7,0:pok
      e s+8,0
105   :
106   rem - loop 83-108 -
107   :
```

```
108     goto 83
109     :
110     if peek(49152)<>76 or peek(49153)<
        >16 goto 83
111     :
112     rem — terug naar desktop —
113     :
114     sys16384
```

** EINDE LISTING tone dial

Checksum Tone dial

REGEL 1	231	REGEL 58	50
REGEL 2	67	REGEL 59	153
REGEL 3	56	REGEL 60	205
REGEL 4	10	REGEL 61	58
REGEL 5	58	REGEL 62	51
REGEL 6	54	REGEL 63	58
REGEL 7	58	REGEL 64	73
REGEL 8	159	REGEL 65	58
REGEL 9	75	REGEL 66	91
REGEL 10	58	REGEL 67	58
REGEL 11	19	REGEL 68	167
REGEL 12	58	REGEL 69	219
REGEL 13	8	REGEL 70	199
REGEL 14	58	REGEL 71	39
REGEL 15	119	REGEL 72	88
REGEL 16	128	REGEL 73	255
REGEL 17	68	REGEL 74	8
REGEL 18	58	REGEL 75	58
REGEL 19	145	REGEL 76	6
REGEL 20	139	REGEL 77	58
REGEL 21	58	REGEL 78	252
REGEL 22	111	REGEL 79	167
REGEL 23	170	REGEL 80	58
REGEL 24	171	REGEL 81	179
REGEL 25	126	REGEL 82	58
REGEL 26	116	REGEL 83	227
REGEL 27	119	REGEL 84	58
REGEL 28	131	REGEL 85	89
REGEL 29	166	REGEL 86	58
REGEL 30	167	REGEL 87	88
REGEL 31	172	REGEL 88	58
REGEL 32	118	REGEL 89	164
REGEL 33	65	REGEL 90	58
REGEL 34	160	REGEL 91	68
REGEL 35	116	REGEL 92	46
REGEL 36	163	REGEL 93	58
REGEL 37	105	REGEL 94	223
REGEL 38	58	REGEL 95	58
REGEL 39	195	REGEL 96	198
REGEL 40	28	REGEL 97	58
REGEL 41	58	REGEL 98	74
REGEL 42	209	REGEL 99	58
REGEL 43	58	REGEL 100	201
REGEL 44	22	REGEL 101	58
REGEL 45	58	REGEL 102	201
REGEL 46	240	REGEL 103	58
REGEL 47	121	REGEL 104	100
REGEL 48	150	REGEL 105	58
REGEL 49	108	REGEL 106	174
REGEL 50	205	REGEL 107	58
REGEL 51	58	REGEL 108	244
REGEL 52	133	REGEL 109	58
REGEL 53	58	REGEL 110	252
REGEL 54	214	REGEL 111	58
REGEL 55	58	REGEL 112	6
REGEL 56	67	REGEL 113	58
REGEL 57	58	REGEL 114	164

Wissen

D. de wijn uit Belgie heeft een korte routine geschreven die het scherm op zo'n leuke manier schoonmaakt dat we U deze toch niet willen onthouden.

```
1       rem wisser - dewijn dieter
2       fora=1024to2023:pokea,100:next
3       forb=1024to2023:pokeb,111:next
4       forc=1024to2023:pokec,121:next
5       ford=1024to2023:poked,248:next
6       fore=1024to2023:pokee,247:next
7       forh=1024to2023:pokeh,227:next
8       forf=1024to2023:pokef,160:next
9       forg=55296to56295:pokeg,6:next
10      print"{SHIFT CLR}"
```

** EINDE LISTING wisser

Checksum Wisser

REGEL 1	23	REGEL 6	69
REGEL 2	49	REGEL 7	73
REGEL 3	53	REGEL 8	65
REGEL 4	56	REGEL 9	106
REGEL 5	68	REGEL 10	112

Ruimterace

n uit Almelo dacht, ach schiet/race spelletjes zijn nooit weg, en stuurde er een aantal in. Hierbij plaatsen we er één van. Elke uitleg is eigenlijk overbodig, het wijst zichzelf dus, veel succes.

```
10      print"[SHIFT-CLR]":poke53280,0:pok
        e53281,0
20      print"[25xCRSR-DOWN]"
30      print"[CTRL-2][4xSPACE]QQ[3xSPACE]
        QQ[3xSPACE]QQQ[3xSPACE]QQQQQQ[2xSP
        ACE]QQQQQQQ
40      print"[COM-8][4xSPACE]QQ[3xSPACE]Q
        Q[2xSPACE]QQQQQ[2xSPACE]QQQQQQQ[SP
        ACE]QQQQQQQ
50      print"[COM-5][4xSPACE]QQ[6xSPACE]Q
        Q[3xSPACE]QQ[SPACE]QQ[3xSPACE]QQ
60      print"[COM-4][4xSPACE]QQQQ[SPACE]Q
        Q[SPACE]QQ[3xSPACE]QQ[SPACE]QQ[SPA
        CE]Q[SPACE]QQ[SPACE]QQQQQQQ
70      print"[COM-5][4xSPACE]QQQQ[SPACE]Q
        Q[SPACE]QQ[3xSPACE]QQ[SPACE]QQ[SPA
        CE]Q[SPACE]QQ[SPACE]QQQ
80      print"[COM-8][4xSPACE]QQ[3xSPACE]Q
        Q[2xSPACE]QQQQQ[2xSPACE]QQ[SPACE]Q
        [SPACE]QQ[SPACE]QQQQQQQ
90      print"[CTRL-2][4xSPACE]QQ[3xSPACE]
        QQ[3xSPACE]QQQ[3xSPACE]QQ[SPACE]Q[
        SPACE]QQ[SPACE]QQQQQQQ
100     print"[COM-6][CRSR-DOWN][18xSPACE]soft
110     print"[6xCRSR-DOWN]
120     foroo=1to4000:next
130     forll=1to25
140     print"[CRSR-DOWN]"
150     nextll
155     goto780
160     tt=0:t=0:gg=0:print"[SHIFT-CLR]"
170     printchr$(14)
180     print"[CTRL-2][SPACE]*************
        *************************
190     print"[4xSPACE]Ruimte[2xSPACE]Race
        ";:print"[2xSPACE]SC=";sc;:print"[
        SPACE]HI=";hi
200     print"[SPACE]********************
        ****************
210     print"[CTRL-8][CRSR-DOWN][11xSPACE
        ]Keuze[SPACE]uit"
220     print"[11xSPACE][9xCOM-Y]"
230     print"[CTRL-4]-Beweeg[SPACE]joysti
        ck[SPACE]naar[SPACE]BOVEN[SPACE]vo
        or[SPACE]een[SPACE]gem[SPACE]akeli
        jk[SPACE]spel.
240     print"-Beweeg[SPACE]joystick[SPACE
        ]naar[SPACE]LINKS[SPACE]voor[SPACE
        ]een[SPACE]nor[SPACE]maal[SPACE]spel."
250     print"-Beweeg[SPACE]joystick[SPACE
        ]naar[SPACE]RECHTS[SPACE]voor[SPAC
```

```
    E]een[SPACE]ex[SPACE]pert[SPACE]spel."
260 print"-BEweeg[SPACE]joystick[SPACE
    ]naar[SPACE]BENEDEN[SPACE]voor[SPA
    CE]HI=0"
270 print"-Druk[SPACE]op[SPACE]FIRE[SP
    ACE]voor[SPACE]stoppen."
280 print"[COM-1][CRSR-DOWN][3xSPACE]*
    *[SPACE]De[SPACE]besturing[SPACE]s
    pel[SPACE]**"
290 print"[3xSPACE]**[SPACE]Joystick[S
    PACE]poort[SPACE]2[SPACE]**"
300 print"[CTRL-1][2xCRSR-DOWN][COM-7]"
310 fora=1to1000:next
320 p=peek(56320)
330 ifp=126thena=a-40:sc=0:gg=gg+1:pri
    nt"[SHIFT-CLR]":goto430
340 ifp=125thena=a+40:hi=0:goto160
350 ifp=123thena=a-1:sc=0:gg=gg+2:prin
    t"[SHIFT-CLR]":goto430
360 ifp=119thena=a+1:sc=0:gg=gg+3:prin
    t"[SHIFT-CLR]":goto430
370 ifp=111thengoto960
380 goto320
390 hi=0:goto160
400 rem *****************************
410 rem ****          spel        ****
420 rem *****************************
430 print"[SHIFT-CLR]"
440 foro=1to300:poke54296,tt
450 z=1244:poke1984,42:poke2023,42:pok
    e56256,10:poke56295,10
460 gosub590
470 ifpeek(z+x)=42thent=t+1:gosub710
480 j=peek(56320)
490 ifj=119thenx=x+1
500 ifj=123thenx=x-1
510 pokez+x,160
520 poke55516+x,7
530 sc=sc+2:tt=tt+1
540 iftt=>15thentt=15
550 nexto
560 gg=gg+1
570 ifgg=4then900
580 goto440
590 ifgg=1then620
600 ifgg=2then640
610 ifgg=3then660
620 q=int(rnd(1)*38):printtab(q)"[CTRL
    2][CRSR-RIGHT]*"
625 q=int(rnd(1)*38):printtab(q)"[CTRL
    2][CRSR-UP][CRSR-RIGHT]*"
630 return
640 q=int(rnd(1)*37):printtab(q)"[CTRL
    2][CRSR-RIGHT]**"
645 q=int(rnd(1)*37):printtab(q)"[CTRL
    2][CRSR-UP][CRSR-RIGHT]**"
650 return
660 q=int(rnd(1)*36):printtab(q)"[CTRL
    7][CRSR-RIGHT]***"
665 q=int(rnd(1)*36):printtab(q)"[CTRL
    7][CRSR-UP][CRSR-RIGHT]***"
670 return
680 rem *****************************
690 rem ***        geluid         ***
700 rem *****************************
710 tt=0:poke53281,2:poke53280,0
720 v=54296:w=54276:a=54277:h=54273:l=
    54272
730 forx=15to0step-1:pokev,x:pokew,129
    :pokea,15:pokeh,40:pokel,200:next
740 pokew,0:pokea,0
750 poke53281,0:ifsc>hithenhi=sc
760 ift=3then160
770 return
780 forl=54272to54296:pokel,0:next
790 poke54296,0
800 poke54277,190:poke54278,248
810 poke54284,190:poke54285,248
820 poke54291,190:poke54292,248
830 s=54276
840 pokes,33:pokes+7,33:pokes+14,33
850 pokes-3,5:pokes-4,1
860 pokes+4,5:pokes+3,2
870 pokes+11,5:pokes+10,3
880 goto160
890 rem:
900 forww=1to100
910 print"[HOME][COM-3][11xCRSR-DOWN][
    14xSPACE]bonus[SPACE]1000
920 print"[HOME][11xCRSR-DOWN][39xSPACE]"
930 nextww
940 print"[HOME][25xCRSR-DOWN]"
950 sc=sc+1000:gg=3:goto440
960 printchr$(142):print"[SHIFT-CLR]":
    poke53280,0:poke53281,0
970 print"[CTRL-4][9xCRSR-DOWN][12xSPA
    CE]home[SPACE]soft[SPACE]1988."
```

** EINDE LISTING ruimterace

Checksum Ruimterace

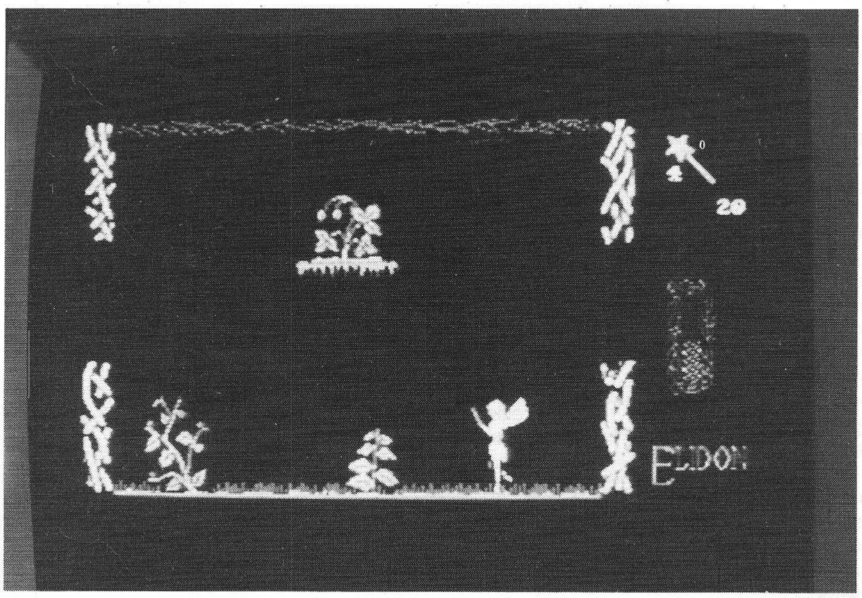
REGEL 10	207	REGEL 530	209
REGEL 20	134	REGEL 540	99
REGEL 30	20	REGEL 550	209
REGEL 40	29	REGEL 560	169
REGEL 50	125	REGEL 570	63
REGEL 60	72	REGEL 580	33
REGEL 70	5	REGEL 590	59
REGEL 80	123	REGEL 600	62
REGEL 90	67	REGEL 610	65
REGEL 100	161	REGEL 620	221
REGEL 110	33	REGEL 625	110
REGEL 120	38	REGEL 630	142
REGEL 130	7	REGEL 640	6
REGEL 140	238	REGEL 645	151
REGEL 150	26	REGEL 650	142
REGEL 155	40	REGEL 660	73
REGEL 160	78	REGEL 665	218
REGEL 170	22	REGEL 670	142
REGEL 180	252	REGEL 680	123
REGEL 190	176	REGEL 690	69
REGEL 200	247	REGEL 700	123
REGEL 210	130	REGEL 710	235
REGEL 220	76	REGEL 720	8
REGEL 230	150	REGEL 730	201
REGEL 240	86	REGEL 740	184
REGEL 250	236	REGEL 750	19
REGEL 260	0	REGEL 760	2
REGEL 270	34	REGEL 770	142
REGEL 280	7	REGEL 780	102
REGEL 290	59	REGEL 790	253
REGEL 300	41	REGEL 800	11
REGEL 310	198	REGEL 810	7
REGEL 320	21	REGEL 820	3
REGEL 330	169	REGEL 830	13
REGEL 340	21	REGEL 840	216
REGEL 350	116	REGEL 850	137
REGEL 360	121	REGEL 860	136
REGEL 370	239	REGEL 870	229
REGEL 380	30	REGEL 880	32
REGEL 390	205	REGEL 890	201
REGEL 400	165	REGEL 900	71
REGEL 410	19	REGEL 910	103
REGEL 420	165	REGEL 920	171
REGEL 430	112	REGEL 930	48
REGEL 440	153	REGEL 940	153
REGEL 450	9	REGEL 950	81
REGEL 460	43	REGEL 960	81
REGEL 470	77	REGEL 970	130
REGEL 480	15		
REGEL 490	6		
REGEL 500	2		
REGEL 510	182		
REGEL 520	2		

Zoals we gewend zijn, heeft Rob Goudriaan weer een aantal games opgedoken, die een paar jaar geleden nieuw waren. Sommige zijn ten onrechte vergeten, andere komen nu weer in de belangstelling.

Oud van Goudriaan

Hidius Bill

Eindelijk weer eens een goed doolhofspel. Er zijn zeer vele liefhebbers voor dit genre. Jammer dat er van dit soort spelen tegenwoordig weinig meer uitkomt. Het is schijnbaar lucratiever om schiet- en simulatiespelen te maken. In dit doolhofspel is het de bedoeling dat je je grote liefde Greta probeert te redden. Voordat je haar in de armen kunt sluiten moet er nog heel wat werk worden verzet. Zo kom je op het eerste level een gigantische hoeveel eieren tegen die allemaal verzameld moeten worden voor je verder kunt. Natuurlijk lukt dit niet zo maar, er zijn altijd wel figuren die je dat proberen te verhinderen. In dit spel lopen er reuzehazen rond. Deze zijn niet alleen enorm groot maar het kost je ook een leven als ze je per ongeluk te pakken krijgen. Gelukkig hoef je niet met lege handen tegen hen in het strijdperk te treden, her en der liggen er wapens. Goed opletten en de juiste middelen pakken om dit eerste niveau te kunnen overleven. Aan de zijkanten zijn er uitgangen. Hoewel..., uitgangen: ga je er links uit, dan kom je rechts weer in en andersom natuurlijk ook. Zijn alle eieren verzameld dan ga je pas door naar het volgende level. Hier blijken er geen wapens tegen de hazen. In het centrum staat een blikje bonen. Dit is absoluut nodig om naar het derde level te kunnen. Om deuren te kunnen openen moeten er verschillende handels omgezet worden anders kun je niet bij het blik bonen komen. De gangpaden moeten weer opgeruimd worden. Zodra alles leeg is ga je naar het derde en laatste level. Hiervandaan kun je je geliefde al zien maar voor je haar in je armen kunt nemen moeten er nog een aantal hindernissen genomen worden. En zoals het hoort, de laatste loodjes wegen het zwaarst. Maar na alle moeite die je gedaan hebt kun je een vreugdedansje gaan maken.
Dit spel is niet zo heel erg moeilijk voor de zeer gevorderde spelfanaten, maar voor de kleine beginner - en die zijn er tenslotte ook - is dit een aardig spel om te oefenen voor je aan het grotere werk te wagen.



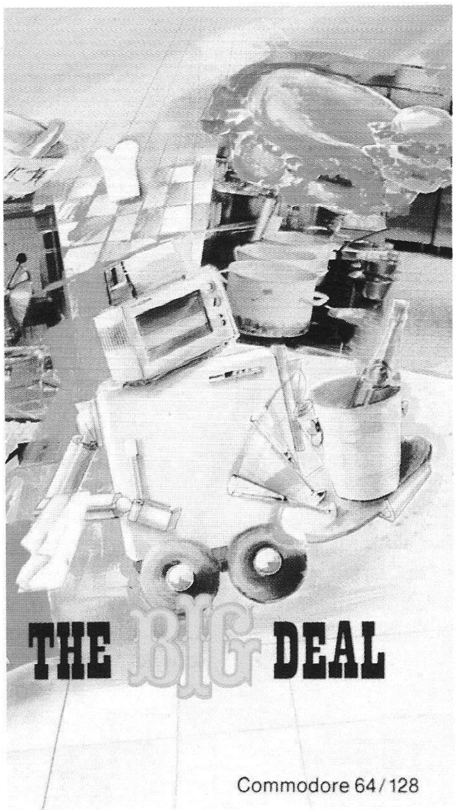
Elidon

Een sprookje, zo kunnen we dit spel uit de oude doos het beste vertalen. Letterlijk maar ook figuurlijk. De hoofdfiguren zijn zo uit het grote sprookjesbos gehaald en neergeplant in dit verhaal.
Ergens diep in het grote bos groeien zeven bloemen van Finvarra. Van deze zeven bloemen moet een kroon gemaakt worden voor de lieftallige elfen-koningin. Iedere bloem heeft een eigen toverdrank nodig om in leven te blijven. Als je dus met het elfje door het onmetelijke bos zweeft weet je wat je te doen staat. Je zult merken dat je pad niet altijd over zachte mospaadjes gaat: achter iedere boom of struik wachten nieuwe gevaren. Enge boomgeesten liggen op de loer om je taak moeilijk te maken, zo niet onmogelijk. Zij willen niets liever dan het bijzondere stuifmeel vernietigen. Pas dus goed op dat je niet te ver het bos ingaat en verdwaald. Het zonlicht kan hier niet doordringen, en het is er dan ook aardedonker. Dit is niet alleen lastig, maar voor kleine elfjes zeer beangstigend. De opdracht lijkt eindeloos lang te duren, maar verlies niet al te gauw de moed. Lukt het na veel proberen eindelijk om je missie te voltooien dan is niet alleen de winter voorgoed verdwenen maar zijn ook de elfjes gered. En zeg nou zelf: wat zou een bos zijn zonder elfjes.

Big Deal

Van onze grootste nederlandse softwaremakers Radarsoft is het spel "Big Deal". Het is een spel dat al enkele jaren geleden op de markt is gebracht, maar nu weer in de belangstelling staat. Ook hebben we hier en daar al een Amiga versie gezien. In het spel zitten twee uitvinders een hamburger zitten te eten. Hierbij ontstaat een plannetje om een computergestuurde robot te maken die alle handelingen kan verrichten die in een snackbar moeten worden uitgevoerd. Dit varieerd van het grillen en frituren, broodjes maken, milksha-

kes en cola inschenken, tot het maken van pizza's. Dit automatiseren kan grote voordelen hebben. De wachttijd voor de klanten zal tot het minnimun beperkt blijven, en de kwaliteit kan altijd goed zijn. De ontwerpers van dit programma hebben ook een naam voor hem bedacht: Floyd heet het prototype en jij mag hem besturen.
Voldoe je aan je opdracht dan worden er ik-weet-niet-hoeveel van deze Floyd's besteld. Het is dus een hele verantwoording die je draagt. Omdat alles moet wennen heb je een week de tijd om je kunnen te tonen, maar eigenlijk moet Floyd dat doen. De werktijden zijn nu niet bepaald vriendelijk te noemen van 9 uur s'morgens tot 9 uur s'avonds, maar een robot wordt natuurlijk niet zo gauw moe.
Voor je aan het echte werk begint inspekteer je eerst de keuken. Is alles oké, dan kan je beginnen. Zo lang er geen klanten zijn kun je het beste vast de recepten bekijken. Alles staat op nette lijsten die kan je oproepen. Maar al snel staat de eerste klant in de zaak en bestelt een broodje rosbief. Dat lijkt niet al te moeilijk: eerst een box pakken uit de kast, een broodje uit de koelkast, de rosbief even bewerken, alles in de box en deze op de lopende band. Zo, de eerste opdracht is voltooid, en als alles zo makkelijk gaat...
Er blijkt alweer een nieuwe klant te wachten. Deze wil een milkshake, dus vlug een glas en de milkshake tappen en hup op de band. Een stuk moeilijker wordt het als er een derde klant een broodje gezond wil en een glas cola. Weer een nieuwe klant wil een big brunch en de vijfde klant komt ook al binnen. Het moet allemaal steeds sneller, en het zweet komt figuurlijk op je voorhoofd te staan. Gelukkig is er voor noodgevallen een voorraadje haute-cuisine tv-dinner, die Floyd in zijn interne opbergkastje heeft staan. Deze is altijd goed om lastige klanten tevreden te stellen, want als je een friet met mayonaise hebt besteld zal je wel je mond houden als je zo iets voorgeschoteld krijgt (en zezeggen dan ook netjes: dank u wel). Duurt het allemaal te lang voor de klanten, dan loop je de kans dat ze met het meubilair gaan gooien. De robot moet dan even gemaakt worden en is dan tijdelijk uit de roulatie. Dit duurt gelukkig niet lang en soms lukt het je ook om de vliegende voorwerpen te ontwijken. Moe, maar voldaan, ben je dolblij als de eerste werkdag er op zit. Maar er wachten er nog een paar. En er word netjes geregistreerd hoeveel tevreden klanten er weg zijn gegaan. Bij de tweede en volgende dagen gaat het "misschien" beter. Maar blijf wel opletten, want niemand vind verbrande frietjes lekker. Gebeurt dit toch, dan kun je het maar beter weggooien. Let wel op als je een bestelling op de band legt, dat het goede nummer op de lichtkrant brand anders is de klant niet tevreden en je weet wat dan de gevolgen zijn: minstens twee ontevreden klanten.
Heeft de zaak aan het eind van de week nog tevreden klanten ? Alles ligt aan jou en een beetje aan onze vriend Floyd.



Big deal



Druid

Druid

Jij, natuurlijk held van zovele spelletjes, de redder in nood, veroveraar van vele planeten, bezweerder van kwade spreuken en nog veel meer van dat. Je moet weer eens aantreden, want er wordt een beroep op je gedaan. Een boze prinses heeft een vloek uitgesproken en overal om je heen huist nu het Grote Kwaad. In de gedaande van een middeleeuwse priester (een Druïde - denk maa aan Asterix) ga je dit kwaad bestrijden. Het spel begint in een verlaten, angstaanjagend landschap. Zo op het eerste gezicht is er niets aan de hand, maar al gauw ontdek je het gevaar. Allerlei geestig gespuis komt op je af als vliegen op een pot stroop. Je staat gelukkig niet geheel met lege handen, want er staan diverse spreuken tot je beschikking, deze kan je afvuren op je vijanden. Ieder plaaggeest moet bestreden worden met zijn eigen wapens. De geesten die in eerste instantie op je afkomen zijn het beste te bestrijden met water. Vliegen zijn allergies voor vuur, etc. Maar helaas heb je geen oneindige voorraad spreuken. Op verschillende plaatsten staan echter kisten, die de broodnodige voorraad aanvullen. Ook zitten er in enkele van deze kisten sleutels. Pak deze altijd want anders kom je niet ver.

Deuren gaan nu eenmaal niet vanzelf open. Ook zijn er spreuken waarbij je onzichtbaar wordt, maar pas op want dat vreet energie. De meest fantastische spreuk is een heel bijzondere "CHAOS". Wat je je hierbij moet voorstellen laat ik graag aan je eigen fantasie over, maar het is wel zeer handig als je even paniek wilt zaaien. Op enkele plaatsen zijn er energiebronnen. Hier is de energievoorraad weer op pijl te brengen. Wat de juiste route is, is moeilijk te zeggen. Uitproberen is de beste remedie (net zo als welke tegenstander bij welke spreuk hoort). Het snel veranderen van de spreuken is dan wel een vereiste. Lastige tegenstanders zijn de Geest, de Slang, het Geraamte en de Duivel.
Als toverspreuken heb je vuur, water en electriciteit. Kijk in iedere kist en haal er uit wat je nodig hebt. Maar pak niet meer dan noodzakelijk, want de teller gaat niet hoger dan 99. Kom je tot het einde van het spel - wat helemaal niet mee zal vallen - dan krijg je een titel mee. De meest geroutineerde speler zal het tot heerser over het licht brengen, een zeer goede gemiddelde speler tot geestenbezweerder of magiër. Maar wat ook al knap is, is het gewoon uitspelen van dit spel. Al is de titel van 'half wit' oftewel 'halve gare' nu niet bepaald complimenteus te noemen. laat je hierdoor niet weerhouden om dit spel te spelen want eenmaal begonnen kom je al gauw in de ban van de "**DRUID**".

The way of the Exploding West

Voor de ware karateliefhebbers, en zij die van aktie houden, is er een programma waar ze hun tanden in kunnen zetten. 'The way of exploding west' is een spel met realistische beelden en een nog realistischer geluid. Jij bent één van de karatespelers, en je kunt tegen een tegenstander of tegen de computer spelen. De bedoeling is natuurlijk om je tegenstander te verslaan. Om te kunnen winnen moet je een hoger aantal punten halen dan je tegenstander. Dit doe je door een goede serie slagen en schoppen. Hiervoor krijg je dan een bepaalde hoeveelheid punten. Als je twee rondes hebt overleefd ga je naar een volgende "**DAN**". Er zijn vier Dan's te verdienen, oplopend in zwaarte. De vierde Dan halen is niet onmogelijk maar zeker ook niet makkelijk. Speel je tegen de computer, dan speelt de tijd een belangrijke rol. Er zijn diverse stoten uit te voeren. Zo zijn er o.a. de flying kick en een voorwaartse salto met achterwaartse trap. Alles speelt zich af op verschillende scenes. Allereerst is er de kloostertuin. Het tweede gevecht vindt plaats op het strand. Overleef je ook dit gevecht dan vervolg je het spel in de sportschool. Als laatste heb je de eer om op de binnenplaats van de tempel te vechten maar dit is niet voor iedereen weggelegd. Een enkeling zal later kunnen vertellen dat hij zijn eer op deze beroemde binnenplaats heeft kunnen verdedigen. Win je ook dit gevecht dan wacht een leuke verrassing. Welke? Ja, dat ontdek je zelf maar. Speel het spel maar uit om daar achter te komen.

Een van de meest fascinerende aspecten van de Commodore 64, zijn de graphics. In de cursus Graphics zullen Michel de Boer en Hylke Sprangers stap voor stap de verschillende grafische mogelijkheden uitleggen. Deze aflevering gaat over karakters.

Graphics op de 64

Deel 1: de karakters

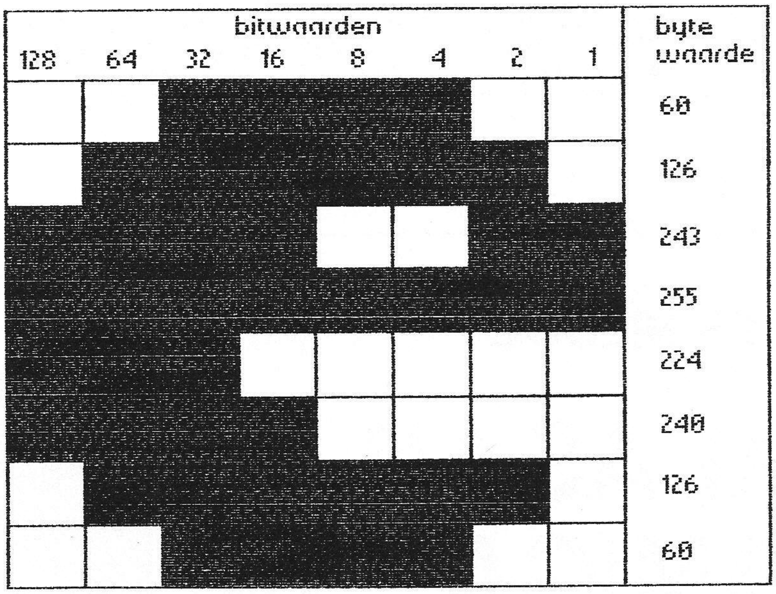
De Commodore 64 kent drie verschillende grafische objecten, te weten: karakters, sprites en high-resolution. In deze cursus zullen deze onderwerpen een voor een aan de beurt komen. Elke aflevering zal eenvoudig beginnen, zodat de beginnende programmeur uit de voeten kan met het behandelde onderwerp. Aan het eind van elk artikel zal wat dieper op de stof worden ingegaan, zodat ook de gevorderde programmeur aan zijn trekken komt.

In deze aflevering zullen we het hebben over karakters. Dit zijn de kleinste van de drie grafische objecten. Ze worden meestal gebruikt om tekst op het scherm te zetten. Door het herdefinieren van de karakters kunnen echter ook mooie achtergrond graphics, bijvoorbeeld voor videospelletjes, worden gemaakt. De verschillende onderwerpen die aan bod zullen komen zijn de volgende:

° Karakters op het scherm zetten
° Opbouw van karakters
° Kleuren
° Zelf karakters maken
° Animatie

Kleur	CHR$	POKE-waarde
zwart	144	0
wit	5	1
rood	28	2
cyaan	159	3
paars	156	4
groen	30	5
blauw	31	6
geel	158	7
oranje	129	8
bruin	149	9
licht rood	150	10
grijs 1	151	11
grijs 2	152	12
licht groen	153	13
licht blauw	154	14
grijs 3	155	15

Figuur 1



Figuur 2

de opbouw van een letter

Karakters op het scherm zetten

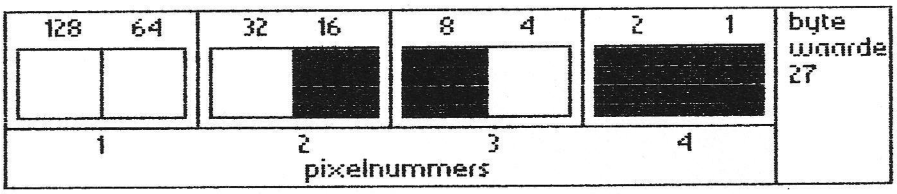
In de Commodore zitten standaard twee voorgedefinieerde karaktersets. Beide sets bestaan elk uit 256 karakters. Een aantal hiervan zijn echter dubbel. Het is helaas niet mogelijk om beide sets tegelijk te gebruiken. Door SHIFT/COMMODORE in te drukken kunt u wisselen van set.
Er zijn twee manieren om deze karakters op het scherm te zetten. De eenvoudigste is door middel van de PRINT instructie. Deze manier behoeft geen verdere uitleg. De tweede manier gaat met behulp van de POKE instructie. Hiervoor moeten we echter eerst meer weten over de schermopbouw. Het scherm bestaat uit 25 regels van elk 40 karakters lang. In totaal staan er dus 1000 (25 * 40) karakters op het beeld (een spatie is ook een karakter). Elk karakter op het beeld correspondeert intern met een geheugenadres. Deze 1000 adressen samen worden het schermgeheugen genoemd. Dit geheugen begint op adres 1024 en eindigt op adres 2023. Het geheugenadres van een bepaalde plaats op het scherm kan nu eenvoudig worden gevonden met de volgende formule: adres = regelnummer * 40 + kolomnummer + 1024. De bovenste regel heeft regelnummer 0 en de linkerkolom kolomnummer 0.
Door nu in deze adressen de schermcodes van de karakters (zie handlei-

ding) te poken, kunt u de karakters op het beeld zetten. POKE 1024,2 zorgt er dus voor dat er linksboven op het scherm een 'B' verschijnt. (Op een aantal oudere Commodores werkt dit niet. Bij deze computers moet behalve de schermcode ook de kleurcode gepoked worden.)
De karakters kunnen in 16 verschillende kleuren op het scherm gezet worden. Bij de PRINT instructie kunt u de kleur gewoon tussen de aanhalingstekens meegeven. Bijvoorbeeld PRINT "{CTRL/BLK} tekst" zorgt ervoor dat er zwarte tekst op het scherm komt. Ook kan de kleur worden veranderd met de instructie CHR$ (zie voor de CHR$-waarde van elke kleur figuur 1). Bijvoorbeeld PRINT CHR$(5) zorgt ervoor dat de karakters, die na deze instructie geprint worden, wit zijn. Een andere manier is weer met POKE. Elk karakter op het scherm heeft, behalve een adres in het schermgeheugen ook nog een adres in het kleurgeheugen, waar de kleur van het karakter in staat. Dit kleurengeheugen loopt van 55296 tot 56295. De kleuradressen worden op dezelfde manier berekend als de schermadressen. POKE 55296,1 maakt het karakter linksboven wit (zie voor de poke-waarden figuur 1).
De scherm- en de randkleur kunnen worden veranderd door in de adressen 53281 en 53280 de betreffende kleur te poken.

Opbouw van karakters

Elk karakter is opgebouwd uit een raster van 8 bij 8 punten. Deze punten worden pixels genoemd. Elk van deze pixels kunnen afzonderlijk uit en aan worden gezet. Een pixel wordt intern gerepresenteerd door een bit. Acht bits vormen samen een byte. Een geheugenadres bestaat dus uit acht bits. Elk bit kan afzonderlijk een 0 of een 1 bevatten. De waarde 0 betekent dat het pixel uitstaat en de waarde 1 dat het pixel aanstaat. Op deze manier worden acht horizontale pixels in een byte gezet. Een karakter bestaat zo uit 8 bytes onder elkaar.

Aan de hand van een voorbeeld zullen we uitleggen hoe de byte-waarden van een karakter berekend kunnen worden. In figuur 2 is een karakter (een pac-man) getekend in een matrix van 8 bij 8.
Een byte bestaat zoals gezegd uit 8 bits. Elk van deze bits heeft een eigen waarde. Zo heeft het meest rechtse bit (bit 0) waarde 1 (2 tot de macht 0).



Figuur 3

Het meest linkse bit (bit 7) heeft waarde 128 (2 tot de macht 7). Door nu de waarden van alle bits, die aan staan (corresponderend met de pixels die aan staan) op te tellen, wordt de waarde van een byte verkregen. De waarde van het bovenste byte in figuur 2 is dus 4+8+16+32 is 60. Op dezelfde manier worden de waarden van de overige bytes berekend. U heeft dus aan het eind acht bytewaarden, die specifiek zijn voor dat karakter.
Als u dus zelf een karakter wilt maken, maakt u eerst een raster van 8 bij 8 en tekent u daar het karakter in. Daarna berekent u de waarden van de acht bytes.

Opslaan van karakters

We weten nu hoe we de acht bytewaarden van een karakter kunnen uitrekenen. Het enige dat we nu nog moeten weten is, waar we deze waarden in het geheugen moeten zetten.
Om een karakterset in het geheugen op te slaan zijn 256 x 8 (= 2048) geheugenplaatsen nodig. De twee standaard karaktersets staan op de ROM (Read Only Memory) adressen 53248 tot 57343. ROM-geheugen betekent dat deze adressen alleen uitgelezen kunnen worden (met de instructie PEEK), maar dat er niet in geschreven (POKE) kan worden. De eerste karakterset begint op 53248 en de tweede op 55296. Deze ROM-adressen zijn echter niet zonder meer uit te lezen omdat dit geheugen normaal door de video-chip gebruikt wordt (onder andere voor het kleurgeheugen). Om dit stuk geheugen uit te lezen moet daarom eerst het video geheugen uitgeschakeld worden. Hiervoor dient bit 2 van adres 1 uitgezet te worden. Als u dit meteen zou doen, dan slaat de computer vast. Dit kunt u verhelpen door vantevoren bit 0 van adres 56334 uit te zetten. Zie voor het aan- en uitzetten van bits de paragraaf over werken met bits.
Nadat u dit gedaan heeft, kunt u de karaktersets uitlezen, maar niet veranderen, omdat het ROM-geheugen is (In ROM kan immers niet geschreven worden). Om de karakterset te veranderen, moet deze eerst naar RAM (Random Access Memory) geheugen verplaatst worden. RAM geheugen is het gewone geheugen, waarmee u normaal werkt. Er kan in geschreven en gelezen worden. Het verplaatsen doen we door de ROM-



adressen met de PEEK-instructie uit te lezen en de zo verkregen waarden in het RAM-geheugen te poken. Als dit gebeurt is, moet bit 2 van adres 1 weer worden aangezet en daarna bit 0 van adres 56334. De karakterset kan echter niet overal in het geheugen worden gezet. Daarvoor zijn bepaalde geheugengebieden gereserveerd. Een daarvan loopt van adres 14336 tot 16383. Om het niet te ingewikkeld te maken zullen we ons beperken tot dit geheugengebied. Het startadres van dit gebied (hier 14336) moet worden doorgegeven aan de computer. Dit gebeurt door bits 1, 2 en 3 van adres 53272 aan te zetten. De andere geheugengebieden zullen over een paar afleveringen aan bod komen. Nu deze vervelende dingen achter de rug zijn, kunnen we aan de leuke dingen beginnen. Eerst nog even een programma, dat de karakterset naar geheugenadres 14336 verplaatst.

```
10 rem karakterset verplaatsen
20 poke 56334,peek(56334) and 254

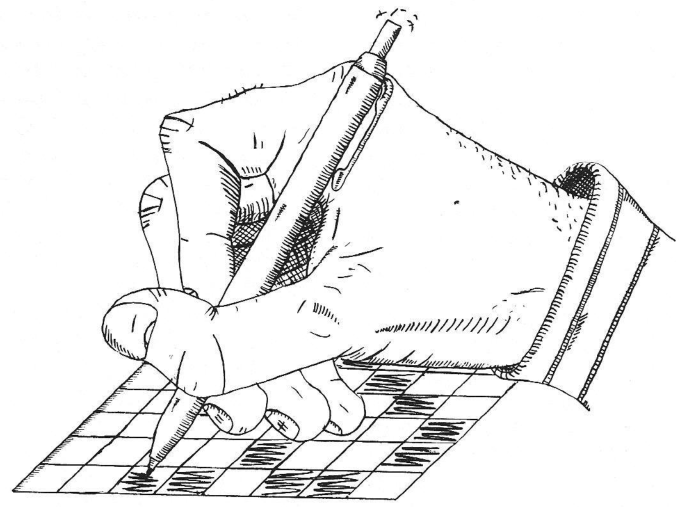
30 poke 1,peek(1) and 251
40 for i = 0 to 2047
50 poke 14336+i,peek(53248+i)
60 next
70 poke 1,peek(1) or 4
80 poke 56334,peek(56334) or 1
90 poke 53272,peek(53272) or 14
```

Karakters veranderen

Nu we de karakterset naar RAM hebben verplaatst kunnen we deze veranderen. Daarvoor moeten we weten waar elk karakter afzonderlijk in de set zit. Dit is heel simpel, want de karakters staan in volgorde van de schermcodes in de set. Zoals eerder gezegd bestaat elk karakter uit acht bytes. Door nu de schermcode van een karakter te vermenigvuldigen met acht en daarbij het startadres van de karakterset op te tellen, wordt het eerste geheugenadres van het betreffende karakter gevonden. Het eerste geheugenadres van bijvoorbeeld de 'A' is 1 * 8 + 14336 is 14344. Dit adres bevat de waarde van het bovenste byte. De volgende zeven adressen (14345 tot 14351) bevatten de rest van de letter 'A'.
Om weer terug te komen op figuur 2; We kunnen nu de 'A' veranderen in een pac-man, door de acht bytewaarden van de pac-man achtereenvolgens te poken in de adressen 14344 tot 14351. Het volgende programma doet dit.

```
10 rem verander 'A' in pac-man
20 for i = 0 to 7
30 read a:poke 14344+i,a
40 next
50 rem byte-waarden
60 data 60,126,243,255,224,240,126,
60
```

kleur hebben; De zogenaamde single color karakters. Er bestaat voor de karakters ook een multi color mode. In deze mode is het mogelijk om een karakter drie kleuren te geven. Dit gaat echter wel ten koste van de resolutie van het karakter. Want het karakter



Plaatjes maken

Met de zojuist verkregen kennis kunnen nu kleine figuurtjes van 8 bij 8 pixels worden gemaakt. Om nu grotere plaatjes, bijvoorbeeld voor achtergronden, te maken, moet het plaatje worden opgesplitst in blokjes van 8 bij 8 pixels. Van elk blokje kan dan een karakter gemaakt worden. Door deze karakters in de juiste formatie op het scherm te zetten, wordt het juiste plaatje verkregen.

Dia-negatieve karakters

De standaard karakterset bestaat uit 256 verschillende karakters. De eerste 128 zijn de karakters die op het toetsenbord zitten. De andere karakters zijn hetzelfde, alleen dan dia-negatief (reverse). Dit betekent dat alle pixels die in het ene karakter aanstaan, in het andere karakter uitstaan en omgekeerd. De schermcodes hiervan lopen van 128 tot 255. Om de schermcode van een bepaald dia-negatief karakter te berekenen, moet er dus 128 bij de schermcode van het gewone karakter worden opgeteld.
De schermcode van een spatie is 32. Dus de reverse spatie heeft als code 160. Dit is het karakter voor de cursor. U kunt nu dus eenvoudig de cursor veranderen.

Multi color karakters

Tot nu toe hebben we alleen nog maar karakters behandeld, die een

bestaat dan nog maar uit een raster van 4 bij 8 pixels in plaats van 8 bij 8. Een pixel in een multi color karakter is twee keer zo breed als een pixel in een single color karakter. Het karakter wordt dus grover.
De pixels worden nu door twee bits in plaats van een bit gerepresenteerd. Deze bits bepalen of het pixel aan of uit staat en de ook kleur van het pixel, als deze aan staat. Met deze twee bits kunnen vier combinaties van bitstanden worden gemaakt (00 01 10 11). De combinatie 00 betekent dat het pixel uitstaat. De overige drie combinaties staan voor de drie kleuren. (het pixel staat dan aan). Een multi color karakter bestaat dus net als een single color karakter uit acht bytes. Alleen correspondeert een byte nu maar met vier pixels; twee bits per pixel. Deze twee pixels liggen steeds naast elkaar in het byte. Bit 0 en 1 corresponderen met het meest rechtse pixel en bit 6 en 7 met het meest linkse.
Helaas is er een beperking bij het kiezen van de kleuren. U kunt namelijk niet elk karakter drie willekeurige kleuren geven. Twee van de drie kleuren moet u voor alle karakters gezamelijk kiezen. De derde kleur kan, net zoals in de single color mode, voor elk karakter afzonderlijk worden gekozen. De eerste en de tweede kleur moeten respectievelijk in de adressen 53282 en 53283 worden gepoked. Deze twee kleuren corresponderen met de bitcombinaties 01 en 10. Zet u bijvoorbeeld de bitcombinatie 10 in bit

3 en 2 van een byte, dan heeft dat tot gevolg dat het tweede pixel van rechts aan staat met de kleur die in adres 53283 staat.
De derde kleur moet weer in het kleuren geheugen (55296 tot 56295) worden gepoked. Voor deze derde kleur geldt ook een beperking. Hiervoor heeft u alleen maar de kleuren met code 0 (zwart) tot en met 7 (geel) tot uw beschikking. Om de goede kleur te verkrijgen moet u bij deze code 8 optellen en dit getal in het kleurengeheugen poken. POKE 55296,8 heeft dus tot gevolg dat het karakter links boven als derde kleur zwart heeft. De reden voor deze beperking is de volgende: U kunt in de multi color mode namelijk ook single color karakters gebruiken. Dit is erg handig om tekst in de multi color mode toch in gewone karakters op het scherm te zetten. Met de derde kleur wordt aangegeven of een karakter single of multi color is. De kleurcodes 0 tot en met 7 geven de single color mode aan en zoals gezegd geven de codes 8 tot en met 15 de multi color mode aan. Deze derde kleur correspondeert met de bitcombinatie 11.
Ter verduidelijking zullen we hier een voorbeeld geven (zie figuur 3). Dit figuur laat een byte van een multi color karakter zien. Het eerste pixel staat hier uit. Het tweede pixel heeft de kleur die in adres 53282 staat (kleur 1). Het derde pixel heeft de kleur die in adres 53283 staat (kleur 2) en het vierde pixel heeft de kleur uit het kleurengeheugen (kleur 3).
We moeten nu alleen nog weten hoe de computer in de multi color mode wordt gezet. Dit kan worden gedaan door bit 4 van adres 53270 aan te zetten: POKE 53270,PEEK(53270) OR 16.

Meerdere achtergrondkleuren

Menu gestuurde programma's kunnen mooi en duidelijk worden gemaakt, door het scherm te verdelen in verschillend gekleurde windows. De commodore biedt de mogelijkheid om elk karakter afzonderlijk een eigen achtergrondkleur te geven. Door blokken tekst op verschillende achtergrondkleuren te zetten, ontstaan windows. Dit is echter alleen maar mogelijk in de single color mode. In de multi color mode, wordt de achtergrondkleur van alle karakters altijd bepaald door de waarde van adres 53281.
Om de computer over te schakelen op het gebruik van meerdere achtergrondkleuren moet bit 6 van adres 53265 worden aangezet. Dit kunt u doen met POKE 53265,PEEK(53265) OR 64.
Het gebruik van meerdere achtergrondkleuren heeft echter wel zijn prijs. U kunt alleen nog maar de karakters met schermcodes 0 tot 63 gebruiken. Deze karakters bestaan uit alle letters, cijfers en leestekens. De karakters die u dus niet kunt gebruiken zijn de grafische karakters. Waarom kunt maar 64 karakters gebruiken? Omdat het mogelijk is om de schermcodes van deze 64 karakters te representeren in zes bits. Het getal 63 ziet er binair als volgt uit 00111111. De eerste twee bits worden dus niet gebruikt. Bij het gebruik van meerdere achtergrondkleuren worden deze twee bits gebruikt om de kleur te bepalen. Net als bij de multi color karakters is het mogelijk om met deze twee bits vier combinaties van

Werken met bits

De Commodore 64 is een acht bits computer. Dit houdt in dat een geheugenadres (een byte) uit acht bits bestaat. Een bit kan aan of uit staan, dat wil zeggen een 1 of een 0 bevatten. Deze acht bits vormen samen het getal, wat in het geheugenadres staat. De waarde van elk bit wordt gegeven door de volgende formule: waarde bit a = 2↑a. Dus bit 0 heeft waarde 2↑0 = 1. De waarde van de geheugenplaats zelf wordt verkregen door de waarden van de bits die aan staan op te tellen
Om bits aan en uit te zetten, moet u eerst weten hoe de logische operatoren AND en OR werken. Met deze twee operatoren kunnen twee getallen bit voor bit met elkaar vergeleken worden. Met de AND en OR worden dus op deze manier twee hele bytes met elkaar vergeleken. De uitkomst van een vergelijking tussen twee bits is zelf ook weer een bit en wordt bepaald door de waarheidstabel van respectievelijk AND en OR. Als twee bits met AND worden vergeleken, is de uitkomst een 1 als beide bits een 1 bevatten, anders is de uitkomst een 0. Bij OR is de uitkomst een 1 als tenminste een van de twee bits een 1 bevat. De volgende twee voorbeeldjes maken het wat duidelijker.

```
byte 1      0 1 0 1 1 0 0 1
byte 2      1 0 0 1 0 0 0 1   OR
            ---------------
uitkomst    1 1 0 1 1 0 0 1

byte 1      0 1 0 1 1 0 0 1
byte 2      1 0 0 1 0 0 0 1   AND
            ---------------
uitkomst    0 0 0 1 0 0 0 1
```

Om nu een bepaald bit in een geheugenadres uit te zetten, kunt u de AND instructie gebruiken. U wilt bijvoorbeeld het tweede bit in geheugenadres 1 uitzetten. De waarde van het tweede bit is 4. De overige bits in geheugenplaats 1 moeten hetzelfde blijven. U kunt het adres vergelijken door middel van AND met 255-4. Het adres wordt zo vergeleken met een byte waarvan alle bits aan staan, behalve het tweede bit. De vergelijking tussen de tweede bits geeft dus altijd 0. Het resultaat zal zijn dat bit 2 op 0 komt te staan en de overige bits hetzelfde blijven. Dit getal moet u dan weer in adres 1 poken: POKE 1,PEEK(1) AND 251.
U kunt ditzelfde bit weer aan zetten met de OR instruktie. POKE 1,PEEK(1) OR 4 heeft tot gevolg dat het tweede bit aan komt te staan en dat de overige bits hetzelfde blijven. Het adres wordt door middel van OR vergeleken met een byte waarvan alleen het tweede bit aanstaat, zodat bij de uitkomst dit bit altijd 1 is. Er kunnen natuurlijk ook een aantal bits tegelijk aan of uit worden gezet. Daarvoor moet u de waarden van de betreffende bits bij elkaar op tellen

```
Bits - AND         Bits - OR
----------         ---------
0 0 - 0            0 0 - 0
0 1 - 0            0 1 - 1
1 0 - 0            1 0 - 1
1 1 - 1            1 1 - 1
```

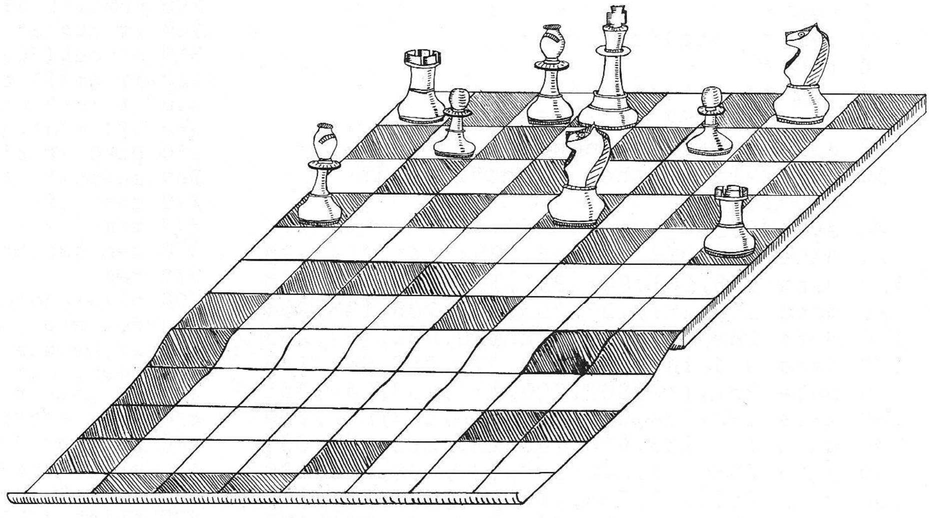
Waarheidstabellen van AND en OR

bitstanden te maken. Er kunnen dus vier verschillende achtergrondkleuren gebruikt worden.
Deze vier kleuren moeten worden gepoked in de adressen 53281 (kleur 1) tot en met 53284 (kleur 4). De bitcombinaties 00, 01, 10 en 11 corresponderen respectievelijk met de achtergrondkleuren die in 53281, 53282, 53283 en 53284 staan.
U kunt karakters met verschillende achtergrondkleuren via het toetsenbord direct op het scherm krijgen. Karakters met kleur 1 verkrijgt u door de karakters gewoon in te tikken. Als u letters samen met SHIFT tikt, krijgt u letters met achtergrondkleur 2. Om cijfers en leestekens in kleur 2 te krijgen moet u op de Commodore toets en een grafische toets drukken. Welke grafische toetsen met welke cijfers en leestekens correponderen kunt u het beste zelf even uitproberen. Om achtergrondkleuren 3 en 4 te gebruiken moet u de computer eerst in reverse mode brengen door middel van CTRL-9. Kleur 3 en 4 worden daarna op dezelfde manier als kleur 1 en 2 verkregen.
Een andere manier om de karakters op het scherm te krijgen is door ze direct in het schermgeheugen te poken. De schermcodes voor de karakters met verschillende achtergrondkleuren kunnen eenvoudig worden berekend met de volgende formule:
Code = gewone schermcode OR kleur,
Hierin heeft 'kleur' de waarde 0 voor kleur 1, 64 voor kleur 2, 128 voor kleur 3 en 192 voor kleur 4.
Het volgende programma geeft een voorbeeld.

```
10 rem meerdere
   achtergrondkleuren
20 poke 53265,peek(53265) or
   64
30 for i = 53280 to 53284
40 read a:poke i,a:next
50 print"commodore info"
60 print"{SHIFT}COMMODORE
   INFO"
70 print"{RVS ON}commodore
   info"
80 print"{RVS ON} {SHIFT}
   COMMODORE INFO"
90 data 0,0,2,5,6
```

Animatie

In een tekenfilm ontstaat beweging door snel een aantal beelden, die een beetje van elkaar verschillen, achter elkaar te projecteren. Op dezelfde manier kunnen we ook karakter graphics animeren. We zullen hieronder kort een aantal methoden bespreken, waarmee u karakter animatie kunt programmeren.



De eenvoudigste animatie is het bewegen van een karakter over het scherm. Door het karakter steeds uit te wissen en op een aangrenzende schermplaats neer te zetten ontstaat de illusie van beweging.
Om realistische beelden te maken is beweging alleen niet voldoende. Vaak zal de vorm van het karakter veranderd moeten worden. Om bijvoorbeeld een mannetje over het scherm te laten lopen, moet de stand van de benen voortdurend veranderd worden. Een snelle methode om dit te doen, is door van te voren van alle standen een apart karakter te maken. Door deze karakters snel over elkaar heen te printen of te poken, lijkt het of het karakter beweegt. Een tweede methode is, door niet alle karakters van te voren te definieren, maar door in het programma steeds hetzelfde karakter te veranderen. Deze methode is minder snel dan de vorige omdat er steeds acht nieuwe waarden in het karaktergeheugen moeten worden gepoked, in plaats van een nieuwe waarde in het schermgeheugen. Ze is daarom ook minder geschikt voor Basic programma's. Een groot voordeel is echter dat er meerdere figuurtjes tegelijk kunnen worden bewogen. Als er bijvoorbeeld tien dezelfde karakters op het scherm staan, zullen deze allemaal tegelijk van vorm veranderen.

Applicatie programma's

Als laatste geven we nog twee leuke, handige programma's. Het eerste programma maakt van uw karakterset italics, schuine letters. Dit is in machinetaal heel makkelijk te doen, door gewoon de bovenste twee bytes van elk karakter ieder een bit naar rechts te schuiven door middel van de LSR-instructie. De onderste drie bytes worden vervolgens met de ASL-instructie ieder een bit naar links geschoven. Dit gebeurt allemaal tijdens het copieren van de karakterset naar RAM. Doordat het programma in machinetaal geschreven is, is het razendsnel. U kunt dergelijke grappen ook zelf inbouwen bij het copieren van een karakterset (in Basic). U kunt bijvoorbeeld van elk karakter de zijkant weghalen, door van elk byte dat gecopieerd wordt, het zesde bit uit te zetten. U zult inmiddels wel weten hoe dit moet (met de AND-instructie). Op een dergelijk wijze kunt u ook onderstreepte karakters maken.
Het tweede programma is een karakter-editor. Hiermee kunt u met behulp van een joystick, een karakter op het scherm tekenen. U kunt bekijken hoe het karakter er uit gaat zien, en als het af is, krijgt u de byte-waarden van het karakter. Met dit programma kunt u ook de opbouw zien van de karakters van de standaard set (met toets '6'). Voor de rest zitten er nog een paar handige opties op. Als u het programma onderbreekt met STOP/RESTORE staat de karakterset nog steeds in het geheugen op adres 14336.

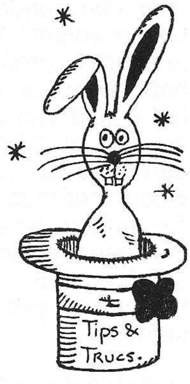
Tot zover deze aflevering over karakters. In de volgende aflevering zullen we alles over sprites vertellen.

Hylke Sprangers en Michel de Boer

```
  1 rem *************
  2 rem *           *
  3 rem *  italics  *
  4 rem *           *
  5 rem *************
 10 for x=49152 to 49255:read a
 20 poke x,a:i=i+a:next
 30 if i<>15383 then print"fout in
   data":end
 40 sys 49152
100 data 173,014,220,041,254,141,014,220
110 data 165,001,041,251,133,001,169,000
120 data 133,251,133,253,169,208,133,254
130 data 169,056,133,252,160,000,177,253
140 data 074,145,251,200,177,253,074,145
150 data 251,162,000,200,177,253,145,251
160 data 232,224,003,208,246,162,000,200
170 data 177,253,010,145,251,232,224,003
180 data 208,245,200,192,000,208,215,230
190 data 252,230,254,165,254,201,216,208
200 data 203,165,001,009,004,133,001,173
210 data 014,220,009,001,141,014,220,173
220 data 024,208,009,014,141,024,208,096
```

```
1 rem *****************
2 rem *               *
3 rem * karakter editor *
4 rem *               *
5 rem *****************
6 rem
7 rem karakterset kopieren
8 rem
10 print"even geduld...":poke 56334,0
20 poke 1,51:for a=0 to 2047
30 poke 14336+a,peek(53248+a):next
40 poke 1,55:poke 56334,1
47 rem
48 rem initialisatie
49 rem
50 poke 53281,254:poke 53280,254
60 restore:xc=1072:qq=32:cu=1
70 u1(1)=86:u1(2)=81:u1(3)=43:u2(1)=32
80 u2(2)=160:u2(3)=32:dim js(16)
90 for x=6 to 15:read a:js(x)=a:next
100 data 41,-39,1,0,39,-41,-1,0,40,-40
107 rem
108 rem karakter invoer
109 rem
110 printchr$(147)chr$(5);
120 input"welk karakter veranderen";a$
130 sc=peek(1050):ga=14336+8*sc
140 print tab(40)"schermcode:"sc
150 print tab(40)"joystick in port 2"
157 rem
158 rem schermopbouw
159 rem
160 for q=0 to 3000:next:print chr$(147)
170 for q=1031 to 1040:pokeq,90
180 poke q+360,90:next
190 for q=1031 to 1399 step 40
200 poke q,90:poke q+9,90:next
210 print tab(100)"karakter : ";a$:print
220 print tab(247)"1 : karakter af"
230 print tab(7)"2 : karakter zien"
240 print tab(7)"3 : scherm leeg"
250 print tab(7)"4 : scherm vol"
260 print tab(7)"5 : reverse karakter"
270 print tab(7)"6 : oude karakter"
277 rem
278 rem joystick & toetsenbord uitlezen
279 rem
280 get a$:poke xc,u1(cu):for t=0 to 80
290 next:if a$="1" or a$="2" then 380
300 if a$>"2" and a$<"7" then 500
310 a=peek(56320):ri=js(abs(a-111))
320 if a=111 then 360
330 if ri=0 or peek(xc+ri)=90 then 280
340 u2(3)=qq:qq=peek(xc+ri):xc=xc+ri
350 poke xc-ri,u2(cu):goto 280
360 cu=cu+1:if cu=4 then cu=1
370 goto 280
377 rem
378 rem karakter omzetten in bytes
379 rem
380 u2(3)=qq:poke xc,u2(cu):a=1072
390 for x=0 to 7:po=0:for t=0 to 7
400 if peek(a+t)=160 then po=po+2↑(7-t)
410 next:poke ga+x,po:a=a+40:next
420 if a$="1" then 450
430 poke 53272,31:for x=0 to 4000:next
440 poke 53272,21:goto 280
450 print chr$(147):for x=0 to 7
460 print"byte"x":";peek(x+ga):next
470 print tab(40)"druk een toets in."
480 get a$:if a$="" then 480
490 clr:goto 50
497 rem
498 rem opties uitvoeren
499 rem
500 a=1072:for x=0 to 7:for t=0 to 7
510 on (asc(a$)-50) gosub 530,540,550,570
520 next:a=a+40:next:goto 280
530 poke a+t,32:return
540 poke a+t,160:return
550 if peek(a+t)=32 then poke(a+t),160:return
560 poke(a+t),32:return
570 if (peek(ga+x) and (2↑(7-t)))=0 then poke
   a+t,32:return
580 poke a+t,160:return
```

Vanaf dit nummer starten Hylke Sprangers en Michel de Boer met de rubriek Tips & Trucs 64. Hierin zullen elke keer handige peeks, pokes en kleine programma's in Basic en machinetaal worden behandeld. Lezers die zelf handige tips en trucs hebben kunnen deze insturen.

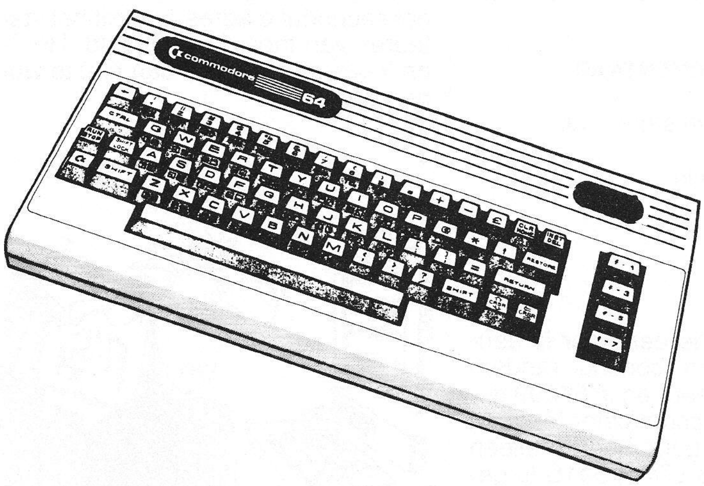


Tips & trucs 64

Toetsenbord buffer

In het geheugen van de commodore 64 is er een zogenaamde toetsenbord buffer. Hierin wordt bijgehouden hoeveel en welke toetsen er door de gebruiker ingetikt zijn, zolang ze nog niet verwerkt zijn. Dit is nodig omdat de computer behalve de invoer van het toetsenbord ook andere gegevens moet verwerken. Tijdens het verwerken van die andere gegevens wordt de invoer van het toetsenbord zolang in deze buffer bewaard, zodat deze later verwerkt kan worden; bijvoorbeeld op het scherm zetten.
Het is voor de programmeur mogelijk om instructies, die pas later moeten worden uitgevoerd, in deze buffer op te slaan. Vanuit een Basic programma kan bijvoorbeeld een instructie, in de vorm van karakters, in de buffer worden geplaatst. Deze instructie wordt pas uitgevoerd, nadat het programma gestopt is.
Deze buffer kan maximaal 10 tekens bevatten en bevindt zich op de geheugenadressen 631 t/m 640. Het aantal tekens dat is opgeslagen wordt bijgehouden in adres 198.
Een handige applicatie is het genereren van Basic-regels vanuit een programma. In onderstaand programma wordt hiervan een eenvoudig voorbeeld gegeven. Hier wordt aan de gebruiker een wiskundige functie gevraagd. Van deze functie wordt dan een programma regel gemaakt, waarna ermee gerekend kan worden.

```
1 rem voorbeeld gebruik
2 rem keyboard buffer
3 rem
10 print"geef
    funktie":input"y= ";a$
20 print"{shift-clr} {2 x
    down}";
30 print"60 def fn y(x)="a$
40 print"goto 60 {home}":poke
    831,13
50 poke 832,13:poke 198,2:end
70 printchr$(147)
80 input"geef x";x
90 fw=fn y(x):print"y="fw
100 print:goto 80
```



In regel 10 wordt de wiskundige funktie gevraagd en in A$ gezet. Nadat in regel 20 het scherm is schoongemaakt, wordt in regel 30 de in te brengen Basic regel op het scherm gezet. Het nummer van deze regel is hier 60 en hierin wordt de ingetypte funktie gedefinieerd met DEF FN. Daarna wordt in de volgende regel GOTO 60 op het scherm gezet waarna de cursor boven aan het scherm wordt geplaatst. Vervolgens worden de eerste twee adressen van het toetsenbord buffer vol gepoked met returns (ASCII-code 13). Adres 198 wordt op 2 gezet (twee returns). Dan stopt het programma vanwege END. Bovenaan het scherm wordt READY neer gezet en de cursor komt eronder te staan, bovenop de Basic regel die geimplementeerd moet worden! De eerste return wordt dan uitgevoerd. Dus die Basic regel komt dan in het programma te staan. De cursor staat daarna een regel verder. Daar staat GOTO 60. Door de tweede return wordt dit ook uitgevoerd en het programma wordt vervolgd bij regel 60, waar de funktie inmiddels staat. De rest van het programma spreekt voor zich.

Let wel: bij het vervolgen van het programma bent u de andere variabelen kwijt! A$ is bijvoorbeeld weg. Dit komt doordat als er iets aan een programma wordt toegevoegd of verbeterd, alle variabelen verloren gaan. Wilt u toch bepaalde variabelen bewaren, dan moet u die ook doorgeven met returns. Dit kan op twee manieren. U kunt de variabelen in DATA-statements zetten, maar u kunt ze ook gewoon in een toekenning op het scherm zetten: A$="var": etc.

Random in machinetaal

Het is erg moeilijk om willekeurige getallen te genereren. Voor een Basic-programmeur is dit geen probleem, omdat hij gebruik kan maken van de geimplementeerde RND functie. De machinetaalprogrammeur moet echter zelf een random functie maken. U zou natuurlijk de ROM-routine, die de basic RND uitvoert, kunnen gebruiken. Deze is alleen nogal langzaam en lastig aan te roepen, omdat er floating point parameters meegegeven moeten worden.
Het kan veel makkelijker en sneller via de geluidschip, al klinkt dit mischien wat vreemd. De golfvorm van ge-

luidsregister 3 kan namelijk worden uitgelezen in adres 54299. Met ruis als golfvorm fungeert dit adres als random generator. De frequentie van de ruis moet hoog worden gekozen voor een goede werking. De ADSR en het volume hoeven niet ingesteld te worden, omdat deze geen invloed hebben op de golfvorm. Verder kan het geluid van stem 3 uitgeschakeld worden door bit 7 van adres 54296 aan te zetten, zodat stem 1 en 2 gewoon voor muziek te gebruiken zijn.
De random getallen die op deze manier gegenereerd worden liggen tussen 0 en 255.
Hieronder volgt een voorbeeld in assembly.

```
MNEMONICS KOMMENTAAR
lda $d418
ora #$80   ;schakel stem 3 uit
sta $d418
lda #$ff   ;frequentie
sta $d40f
lda #$80   ;ruis
sta $d412
rts
```

Nadat deze routine een keer is aangeroepen, worden continu random getallen gegenereerd en in 54299 gezet. Om dus de accumulator te laden met een random getal hoeft u alleen maar de instructie LDA $D41B te geven.

Laden en saven in machinetaal

Voor het laden en saven van files vanuit machinetaal programma's kunt u gebruik maken van ROM-routines. In dit stukje zullen we uitleggen wat de werking van deze routines is en hoe u ze zelf kunt gebruiken.

$FFBD:
Hiermee wordt de filenaam aan de computer doorgegeven. Voor het aanroepen van de routine moet de lengte van de naam in de accumulator worden gezet. Het X- en het Y-register moeten respectievelijk het LO- en HI-byte van het adres waar de naam is opgeslagen, bevatten.

$FFBA:
Met deze routine wordt een file geopend. In de accumulator moet het filenummer staan. Dit nummer kunt u zelf kiezen tussen 0 en 255. Het X-register moet het devicenummer bevatten. Dus 1 voor tape en 8 voor disk. Het Y-register bevat het secundaire adres.
Bij LOAD"FILE",8,1 is het secundaire adres 1 en het devicenummer 8. Bij LOAD"FILE" is het secundaire adres 0 en het device nummer 1. Zie de routines $FFD5 en $FFD8 voor nadere uitleg over dit secundaire adres.

$FFD5:
Dit is de laad/verify routine. Voordat deze routine wordt aangeroepen, moet er een file geopend zijn en een filenaam gegeven zijn met behulp van de vorige twee routines.
Als voor de aanroep de accumulator 0 is, wordt er geladen en bij 1 wordt er geverifieerd. Bij het secundaire adres 0 moeten het X- en Y-register het LO- en HI-byte van het startadres, vanaf waar het programma in het geheugen moet worden geladen, bevatten. Bij het secundaire adres 1 wordt het startadres van tape/disk gehaald. Het X- en Y-register hoeven dan niet te worden gezet.



Na het laden van de file bevatten het X- en Y-register het eindadres van de file.

$FFD8:
Dit is de save routine. Voor de aanroep deze routine moeten weer de routines $FFBA en $FFBD worden aangeroepen. U moet bij het saven altijd het secundaire adres 1 gebruiken.

In twee opeenvolgende zero-page adressen (u kunt hiervoor bijvoorbeeld $FB en $FC gebruiken) moeten het LO- en HI-byte van het start adres van de te saven file gezet worden. De accumulator moet het eerste zero-page adres bevatten (in ons voorbeeld dus $FB). Het X- en Y-register het eindadres.

Het volgende voorbeeldprogramma schrijft het geheugen van $c000 to $cfff naar tape.

```
LABELS  MNEMONICS    KOMMENTAAR
        lda #$01     ;filenr is 1
        ldx #$01     ;datarecorder
        ldy #$01     ;sec. adres 1
        jsr $ffba
        lda #$04     ;naamlengte
        ldx # aam    ;lo-byte naam
        ldy #naam    ;hi-byte naam
        jsr $ffbd
        lda #$00     ;lo-byte
                       start
        sta $fb
        lda #$c0     ;hi-byte
                       start
        sta $fc
        lda #$fb     ;zero-page
                       adr
        ldx #$ff     ;lo-byte eind
        ldy #$cf     ;hi-byte eind
        jsr $ffd8
        rts
naam    .asc "file"
```

Peeks & Pokes

650: Toets repeat
In dit adres wordt bepaald welke toetsen er repeteren. Standaard staat er een 0 in, zodat alleen de spatiebalk, cursortoetsen en de insert/delete toets repeteren. Met POKE 650,128 repeteren alle toetsen. Met POKE 650,64 repeteert geen enkele toets.

808: STOP/RESTORE uitschakelen
Met POKE 808,234 kunt u de werking van de STOP/RESTORE uitschakelen. Helaas werkt het LIST commando dan niet meer goed. Met POKE 808,237 is alles weer normaal.

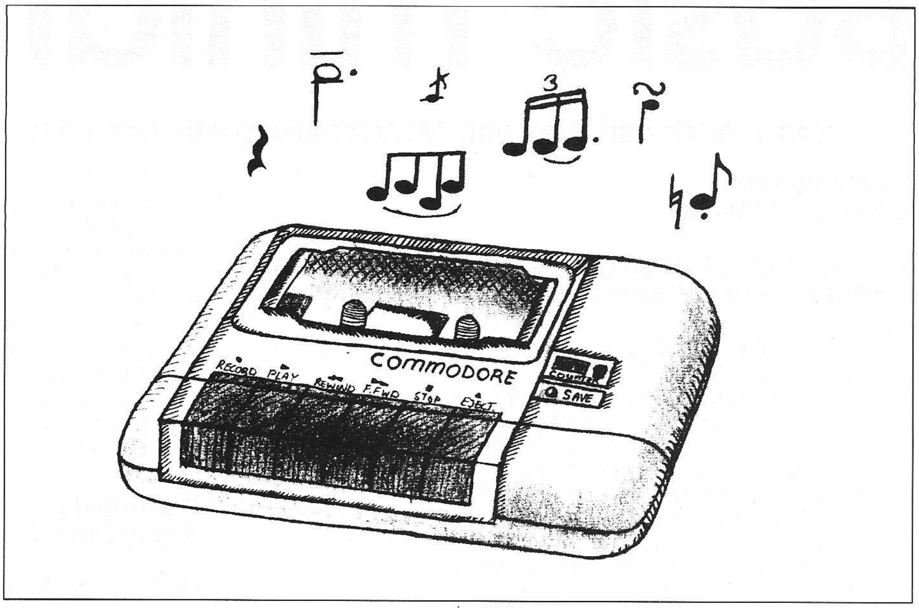
214: Huidige cursor positie
Dit geheugenadres kan worden uitgepeeked om te kijken op welke regel de cursor staat. De bovenste regel heeft regelnummer 1.

653: Shift/CTRL/Logo test
Met dit adres kan getest worden of er op de Shift-, Commodore- of Controltoets wordt gedrukt. De bitverdeling is als volgt:
bit 0: Shift
bit 1: Commodore Toets
bit 2: Control
Als er op de Control wordt gedrukt heeft dit adres dus de waarde 4. Combinaties van toetsen zijn ook mogelijk. Voorbeeld: als u alle drie de toetsen tegelijk indrukt, is de inhoud van dit adres 7

Geluid uit de datarecorder

Het zoeken van een programma, waarvan de tellerstand niet bekend is, is een vervelend en vaak langdurig karwei. Om dit zoeken een beetje te versnellen, kunt u het volgende programma gebruiken.
Het programma zorgt er voor dat u de signalen van de datarecorder kan ho-

ren. Om nu het begin van een programma te vinden moet u steeds een stukje spoelen en een stukje afspelen, totdat u een hoge pieptoon hoort (deze toon is het begin van een programma). Daarna kunt u het programma laden. U hoeft nu niet lang te wachten voordat het programma gevonden wordt. Het op deze manier zoeken van het begin van een programma kost minder tijd, dan gewoon laden en wachten tot de computer het gevonden heeft.




```
10 rem geluid uit de
     datarecorder
15 rem
20 for i = 49152 to 49194
25 read a:b=b+a:poke i,a
30  next
40  if b<>5303 then print"fout
    in data!"
60  data 32,23,248,165,145,
    201,127
65  data 208,1,96,169,0,141,17
70  data 208,173,13,220,41,
    16,74
75  data 141,24,212,141,32,
    208,165
80  data
    197,201,60,208,238,169,0
85  data 133,198,169,27,141,
    17, 208, 96
```

Nadat u dit programma geRUNd heeft, kunt u een tape afluisteren met SYS 49152. Als u klaar bent met zoeken, moet u op de spatiebalk drukken.

Hylke Sprangers en Michel de Boer

Basic miniatuurtjes

Een rubriek van Alex van Maarschalkerwaart met korte tot zeer korte programma's

Zwaardgevecht
Auteur: Dusty Murray

Zwaardgevecht 64 is een geluids demo van twee zwaardvechtende personen het is alleen om naar te luisteren.

```
0   print"[SHIFT-CLR]"tab(11)"zwaardge
    vecht[SPACE]c64"
5   fort=54272to54296:poket,0:next:pok
    e54296,15:a=54270:b=54280
10  pokea+2,200:poke a+3,100:poke a+6,
    32:poke a+7,20:poke a+8,14
20  pokeb,210:pokeb-1,8:pokeb+4,101:po
    keb+5,15:pokeb+3,33:poke a+6,33
40  if int(15*rnd(1))<>1 then goto 40
50  pokeb+3,0:pokea+6,0:poke b+3,33:po
    ke a+6,33:goto40
```

Tekst Scroller
Auteur: Dusty Murray

```
0   fort=49152 to 49212 : read a:poket
    ,a:next
10  a$="[CTRL-9][SPACE]commodore[SPACE
    ]info[SPACE]!!!!!!!!!![SPACE][CTRL
     0][3xSPACE]"
20  print"[CTRL-2][SHIFT-CLR]"a$:fort=
    0to 59:poket+832,peek(t+1024):nex
    t:poke 792,193:sys 49152
100 data 120, 169, 12, 141, 20, 3
110 data 169, 192, 141, 21, 3, 96
120 data 162, 0, 189, 64, 3, 157
130 data 0, 4, 232, 224, 40, 208
140 data 245, 162, 0, 172, 64, 3
150 data 189, 65, 3, 157, 64, 3
160 data 232, 224, 39, 208, 245, 140
170 data 103, 3, 169, 0, 141, 32
180 data 208, 141, 33, 208, 141, 134
190 data 2, 238, 134, 2, 76, 49,234
```

Tekstbalk
Auteur: Dusty Murray

```
5   fort=0to46:read a:poke49152+t,a:ne
    xt:fort=832to832+40:poket,32:next
10  a$="[CTRL-9][5xSPACE][CTRL-A][2xSP
    ACE][CTRL-A][4xSPACE]commodorere[S
    PACE]info[SPACE]!!![3xSPACE][SHIFT
    SPACE][5xSPACE]"
20  print"[CTRL-2][SHIFT-CLR]";a$:fort
    =0to40:poket+832,peek(t+1024):next
    :sys 49152
200 data120,169,12,141,20,3,169,192,14
    1,21,3,96,169,1,141,13
202 data 2, 162, 0, 189, 64, 3, 157, 0
203 data 4, 232, 224, 40, 208, 245, 169, 0
204 data 141,32,208,141,33,208,76,49,2
    34,0,0,0,0,0,0
```

Lijnen
Bij het programma op lijnen moet u als uw een power cartridge heeft die uitschakelen d.m.v quit

```
0   print"[CTRL-G]":poke53265,100:poke
    52,48:poke56,48:clr:a=-1
20  poke56334,peek(56334)and 254:poke
    1,peek(1)and 251:fori=0to2048
40  pokei+12288,peek(i+53248):a=a+1:if
     a=7 then a=-1:pokei+12288,255
41  next
42  poke 1,peek(1)or 4:poke 56334,peek
    (56334) or 1:poke 646,6:print"[SHI
    FT CLR]"
70  poke53280,0:poke53281,1:poke646,6:
    print"[SHIFT-CLR]":poke53265,91:po
    ke 53272,28
```

Bladomsla
Auteur:Dusty Murray

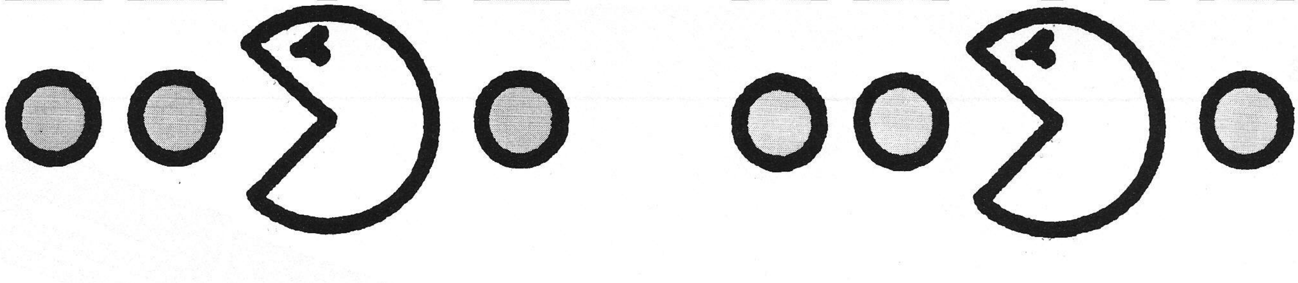
bladzijde is een elegantere manier om het beeld te wissen er wordt als het ware een bladzijde omgeslagen

```
10  c=49152:fort=0to46 :readn:poket+c,n:next
110 data162,255,189,192,6,157,232,6,20 2,208,247
120 data162,255,189,193,5,157,233,5,20
    2,208,247
130 data162,255,189,194,4,157,234,4,20
    2,208,247
140 data162,255,189,195,3,157,235,3,20
    2,224,60,208,245,96
400 fort=1024to1064          :poket,160:
    sys c:next:fort=1to22:sysc:next
410 fort=1024to1064          :poket,32   :
    sys c:next:fort=1to22:sysc:next
420 print"[HOME][5xCRSR-DOWN][12xCRSR-
    RIGHT]commodore[SPACE]info
```

Banden Ontwijken
Auteur: Dusty Murray

Dit is een eenvoudig spel waarbij u een auto bent onderaan het scherm en u moet omhoog rijden en banden ontwijken u bestuurt de auto d.m.v links- (,) rechts- (.)

```
0   print"[CTRL-1][SHIFT-CLR]":a$="W[C
    TRL 9][4xSPACE]B[4xSPACE][CTRL-0]W
    ":sc=0:poke 650,128:poke53280,0:po
    ke53281,8
10  c=49152:fort=0to46 :readn:poket+c,
    n:next:fort=54272to54295:poket,0:n
    ext
110 data162,255,189,192,6,157,232,6,20
    2,208,247
120 data162,255,189,193,5,157,233,5,20
    2,208,247
130 data162,255,189,194,4,157,234,4,20
    2,208,247
140 data162,255,189,195,3,157,235,3,20
    2,224,60,208,245,96
200 a=10:fort=1to10:poke 1064+39*rnd(1
    ),87:sys c:next:poke54296,0:poke54296,15
300 d=a+4
400 sc=sc+1:poke54296,0:poke54296,15
410 print"[HOME][CRSR-DOWN]"tab(37*rnd
    (1))"WW":sys c
431  ifpeek(1942+d)=87orpeek(1943+d)=87
    orpeek(1944+d)=87thengoto 600
432  poke1942+d,81:poke1943+d,160:poke1944+d,81
440  getb$:ifb$=","and d>2then d=d-1
```



```
450 ifb$="."and d<37then d=d+1
500 goto 400
600 print"[CTRL-2][HOME]score:"sc:prin
    t"[CTRL-2][HOME][23xCRSR-DOWN]"tab
    (d-3)"boem!![HOME]"
610 fort=1to16step 2 :poke54296,t:poke
    54296,0:next:poke 54272,20:poke54273,20
615 poke54278,112:poke54277,0 :poke54
    276,128:poke54276,129:poke54296,15
620 fort=16to0step-.08:poke54296,t:pok
    e54296,0         :next
```

Sprite in data

Sprite in data is een programma om sprites, die in het geheugen staan in data regels te zetten het vraagt u 2 dingen 1 welke sprite (0-255) b v sprite 13 staat acher geheugen 832 (64*13=832).

```
0    rem---------------------------------
2    print"[SHIFT-CLR][COM-5][2xSPACE]t
     ype[SPACE]-1[SPACE](return)[SPACE]
     voor[SPACE]terug[SPACE]in[SPACE]ba
     sic":poke53281,0:poke53280,12
4    input"[COM-5][4xCRSR-RIGHT][4xCRSR
     -DOWN]welk[SPACE]sprite[SPACE]numm
     er[SPACE](0/255)";a:a=a*64:ifa<0thenend
6    input"[9xCRSR-DOWN][2xCRSR-RIGHT]n
     a[SPACE]welke[SPACE]regel[SPACE]de
     [SPACE]data[SPACE]setten";dr:ifdr=
     -1thenend
8    ifdr<100thenprint"[10xCRSR-DOWN]??
     ?[SPACE]error[SPACE]:[CTRL-2]";:go
     to 22
10   dr=dr+10:print"[SHIFT-CLR][3xCRSR-
     DOWN]":fort=1to8:a$=""
12   b$=str$(dr):b$=right$(b$,len(b$)-1
     ):a$=b$+"[SPACE]data":b$="":fors=0
     to6
14   b$=str$(peek(a+s)):b$=right$(b$,le
     n(b$)-1):a$=a$+b$+",":b$="":next
16   b$=str$(peek(a+7)):b$=right$(b$,le
     n(b$)-1):a$=a$+b$
18   printa$:dr=dr+10:a=a+8:next:print"
     [CTRL-1]run[SPACE]4[HOME]";
20   poke 198,11:fort=631to631+10:poket
     ,13:next:end
22   print"[CTRL-9]getal[SPACE]is[SPACE
     ]te[SPACE]klein"
24   print"[2xSPACE][CTRL-9]dat[SPACE]z
     ou[SPACE]dit[SPACE]programma[SPACE
     ]overschrijden[CTRL-0][COM-5]"
26   goto 4
28   rem--------------------------------
```

Kleurenscrol

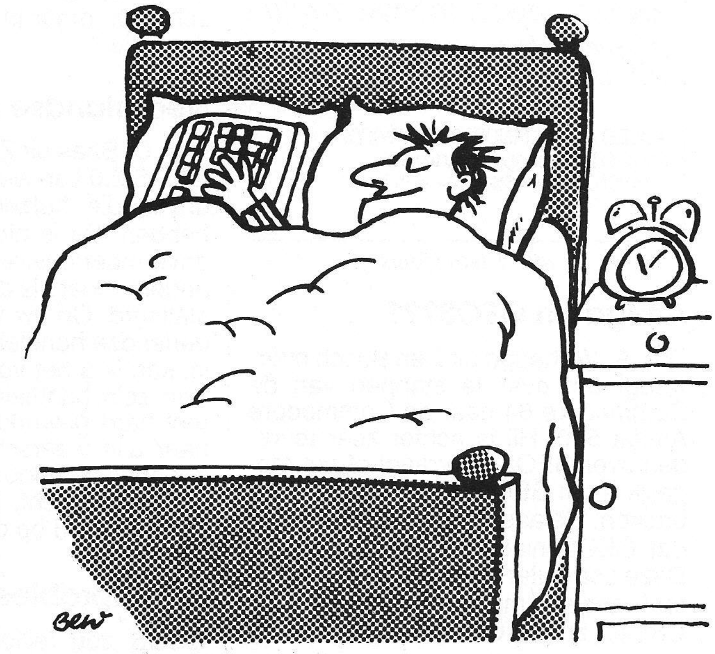
```
10   poke53265,91:poke 53280,0:poke5328
     1,0:print"[COM-1][SHIFT-CLR]";:pok
     e53282,7:poke53283,10
20   a$="[COM-7][CTRL-9][2xSPACE][CTRL-
     7][CTRL-9][2xSHIFT-SPACE]":print"[
     HOME]";:fort=1to11::printa$;:next:
     fort=1064to2023
30   poket,peek(t-40):poke54272+t,peek(
     54232+t):next:poke53284,2
40   print"[HOME][11xCRSR-DOWN]":fort=1
     to3:printtab(9)"[22xSHIFT-SPACE]":next
45   print"[HOME][13xCRSR-DOWN]"tab(11)
     "[CTRL-2][2xSPACE]commodore[SPACE]
     info[2xSPACE]"
50   poke53265,91:::poke53265,27::::goto 50
```

Audio

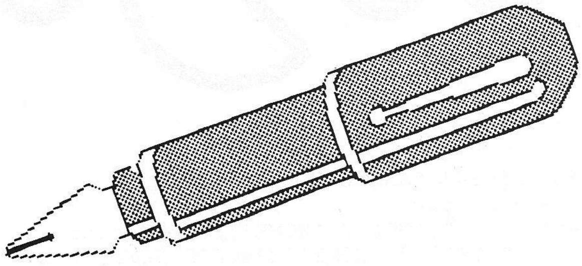
Auteur: Dusty Murray

Audio is een programma om het bijvoorbeeld het begin van een programmatje op een bandje te vinden het werkt simpel :
Run het programmatje ,stop een bandje in de datasette en houdt de funktie toets f-1 ingedrukt dan druk play op tape en wacht ondertussen f-1 gewoon inhouden als u genoeg heeft gehoord laat u f1 los u zit dan weer in basic als u weer wat noddig heeft van bandje en u heeft al runstop restore gedaan typ dan sys 49152 (return) en gebruik f-1 weer om na de cassette te luisteren u kunt na run gewoon new intypen als uw een ander basic programma heeft ingeladen en gerund dan werkt het programma nog steeds gewoon door het programma heen.

```
5    b=0:fort=0to 60:reada:b=b+a:poke
     49152+t,a:next:if b =7324thensys 49152:end
10   print"[CTRL-2]fout[SPACE]in[SPACE]data!!!!"
200 data 120,169,12,141,20,3,169,192
201 data 141,21,3,96,166,197,224,4
202 data 240,3,76,49,234,174,32,208
203 data 142,64,3,169,0,141,32,208
204 data 162,0,160,0, 173, 13, 220, 74
205 data 141,32,208,141,24,212,232,208
206 data 243,200,208,240,173,64, 3, 141
207 data 32,208,76,49,234
```



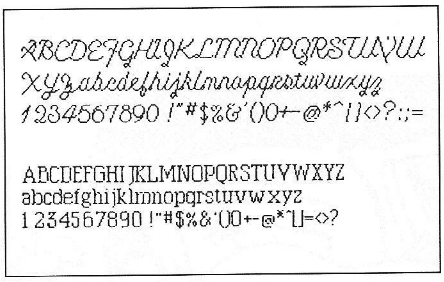
Geos - INFO



Van dhr. Redering uit Ridderkerk ontvingen wij een brief met de mededeling dat zijn GEOS 2.0 niet naar behoren functioneert. Vermoedelijk is er iets fout gegaan bij het installeren. Hieraan kunnen wij echter weinig doen. Dit is een probleem voor de leverancier. In uw geval kunt u zich het beste wenden tot de importeur van GEOS 2.0, FROTECH Nederland BV, postbus 189, 3620 AD Breukelen.

Softwareproblemen

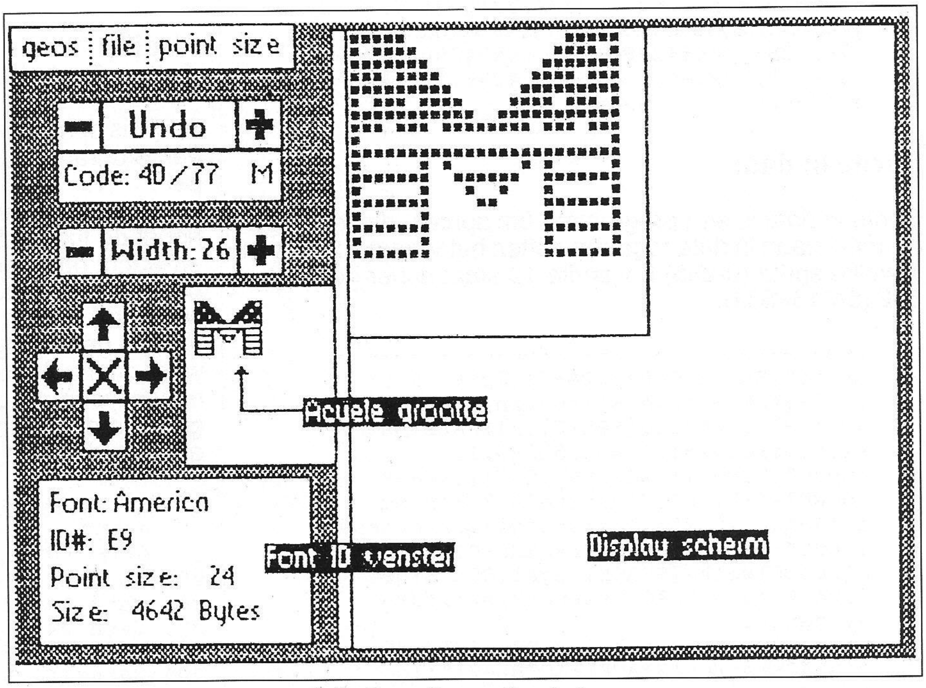
Ook voor dhr. Weenink uit Borculo, die problemen heeft met GeoPublish, geldt dat hij contact op moet nemen met zijn leverancier. Verder is uit uw brief moeilijk op te maken wat het exacte probleem is. Daarom een verzoek van ons aan een ieder, als u een vraag of probleem heeft, beschrijf deze dan duidelijk. Korte, zakelijke en vooral duidelijke brieven kunnen wij in behandeling nemen. En voor softwareproblemen, zoals het niet goed functioneren van een programma, geldt dat u zich moet wenden tot de leverancier en niet tot onze rubriek.



afb. 1 letterfonts

Amiga en GEOS???

Dhr. A. Verheugd uit Den Bosch overweegt om over te stappen van de Commodore 64 naar de Commodore Amiga 500. Hij is echter zeer tevreden over GEOS en vraagt of het mogelijk is om GEOS op de Amiga te gebruiken. Helaas moeten wij u melden dat GEOS niet werkt op de Amiga. Deze computer heeft namelijk een totaal andere microprocessor als de C64 en zal de GEOS machinetaalinstructies dus niet begrijpen. We kunnen u voorts meedelen dat conversie soft- en hardware hiervoor ook niet verkrijgbaar is. Of GEOS ooit voor de Amiga op de markt komt is onwaarschijnlijk, maar je weet maar nooit in deze tijd.



afb. 2 geoFont edit-window

Nederlandse handleidingen

Dhr. C. Baas uit Zwijndrecht vraagt of GEOS 2.0 kan werken met twee diskdrives die hetzelfde devicenummer hebben. Dit is niet mogelijk. De ene drive moet devicenr. 8 hebben en de andere moet als devicenr. 9 zijn geïnstalleerd. Op uw vraag of er een Nederlandse handleiding van GEOS 2.0 is, kan ik u het volgende mededelen. Aan zo'n handleiding wordt momenteel hard gewerkt. Medio juni zal er naar alle waarschijnlijkheid een Nederlands handboek GEOS 2.0 worden uitgebracht. We houden u hiervan uiteraard op de hoogte.

Printerprobleem!!

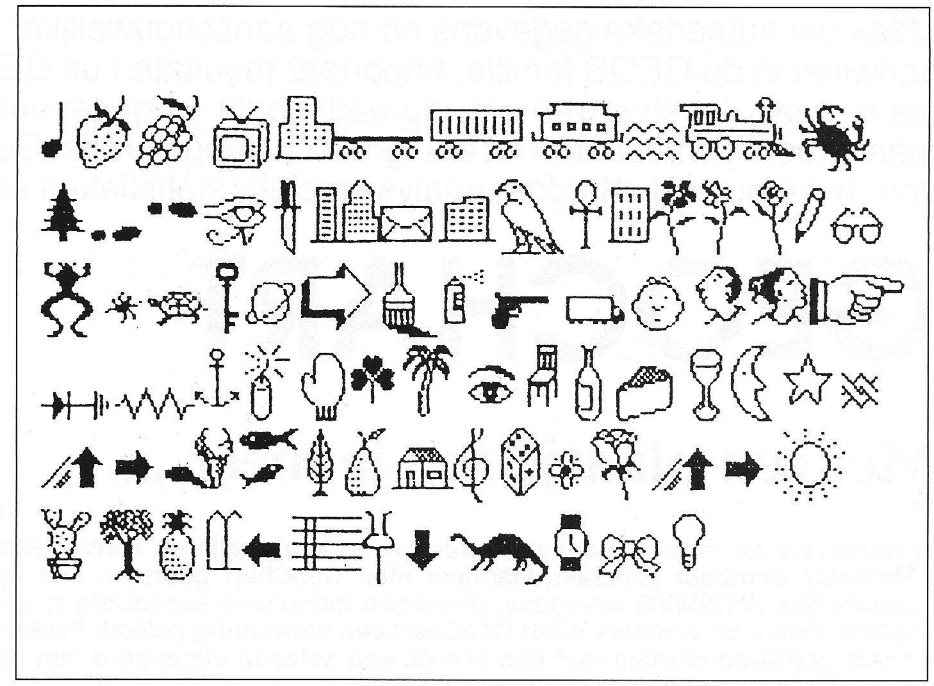
GEOS zou feilloos moeten werken met een STAR NX-10C printer, als het versie 2.0 betreft. Maar dhr. H. Reusink uit Amstelveen beschikt echter alleen over GEOS 1.3. Misschien zou u een andere printerdriver (bv. STAR SG-10C) kunnen proberen. Als dit echter niet het beoogde resultaat oplevert blijft alleen de aanschaf van GEOS 2.0 over om het probleem op te lossen. Het is weliswaar niet de meest charmante oplossing, maar misschien wel de enige.

FontPack Plus

Het is in Amerika al een tijdje verkrijgbaar, maar nu ook in Nederland, **FontPack Plus**. De opvolger van FontPack 1 bezit maar liefst 53 verschillende lettertypes, oftewel letterfonts, die u kunt gebruiken met GEOS op de C64 en C128. Naast een flink aantal letters en karakters, zijn er ook symbolen op de diskette aanwezig. Dit zijn o.a. Ashby 23, Barrington 16/32, Hillgord I 28 en Hillgord II 28, Laurence 12, McLauglin 18 en Wurster 25. Voor mensen die braille willen leren is er Braille 12/24.

En mocht u aan deze hoeveelheid fonts nog niet genoeg hebben dan kunt u met de meegeleverde applicatie GeoFont, zelf letterfonts ontwerpen. Of wanneer u een bestaande font wilt wijzigen. Hierbij kunt u denken aan het aanpassen van enkele karakters, die u met een leesteken wilt uitrusten, zoals ï, é, ë, enz. De fonts zijn gemakkelijk te ontwerpen, met de zgn. pixel-edit modus. Blokje voor blokje bouwt u een letter op en het resultaat is steeds op ware grootte op het scherm zichtbaar.

De grootte van de font wordt steeds in **points** en in **bytes** weergegeven. Ook het wijzigen van het **ID** (identificatie) van een font is mogelijk. Een handig programma om zelf met letterfonts te experimenteren. FontPack Plus kost momenteel fl. 149,00.



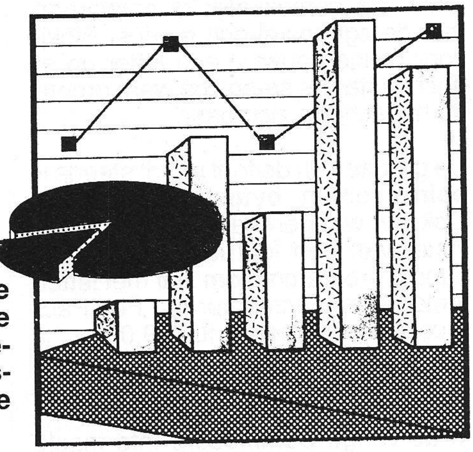
Verschillende fonts van Fontpack Plus

GEOS Gebruikers Groep

In de vorige Commodore Info maakten we al melding van de fa. Computrans uit Almere. Computrans was op het gebied van GEOS zeer aktief. Deze aktiviteiten zijn nu zo positief ontvangen, dat men heeft besloten de aktiviteiten nog verder uit te breiden. Dit houdt ondermeer in dat er een Geos Gebruikers Groep is opgericht, die zich ten doel stelt het gebruik van GEOS nog meer bekendheid te geven. Tevens verzorgt deze Geos GG ondersteuning bij GEOS applicaties.De uitgave van een nieuwsbulletin is tevens één van de zaken die door de Geos GG worden gerealiseerd. Verder behoort het geven van van cursussen in GEOS tot de werkzaamheden van deze gebruikersgroep. Voor meer informatie kunt u zich wenden tot **Geos Gebruikers Groep,** postbus 52, 1300 AB Almere.

Maak uw numerieke gegevens nu nog aanschouwelijker met GeoChart, de nieuwste aanwinst in de GEOS familie. Importeer resultaten uit GEOS georiënteerde programma's, zoals tekstverwerkers, spreadsheets of gegevensbestanden. Maak op deze manier bedrijfsresultaten tot een grafisch hoogstandje. GeoChart is bij uitstek geschikt voor huishoudelijke budget analyses, club- statistieken en "what if" projecties.

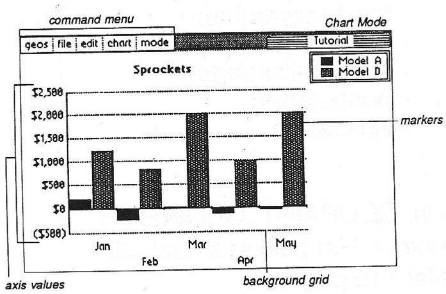
GEOCHART



Met een plaatje zeg je meer.....

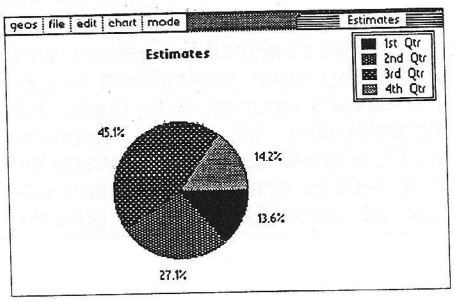
Numerieke en virtuele gegevens worden op een snelle en eenvoudige manier zichtbaar gemaakt, wanneer men GeoChart gebruikt. Met de gebruikelijke WYZIWYG weergave, pull-down menu's en eenvoudig te gebruiken iconen en vensters biedt GeoChart een volwaardig pakket. Professionele grafieken worden met één klik en een selectie uitgevoerd met de snelheid, die u van GEOS bent gewend.

Het heeft even geduurd, maar nu is het er eindelijk, een applicatie om de cijfertjes uit bv. GeoCalc om te zetten in een goedogende grafiek. Voordat GeoChart op de markt verscheen, was het evenwel ook goed mogelijk om grafieken te vervaardigen met één van de GEOS applicaties, namelijk GeoPaint. Probleem hierbij was echter dat de verhoudingen tussen de verschillende delen niet altijd overeenstemden met de bijbehorende getallen. Dit werd voornamelijk veroorzaakt omdat men zelf de delen moest tekenen. Dit behoort, met de komst van GeoChart, tot het verleden. Het pakket bevat maar liefst 9 verschillende "chart" diagrammen, waaronder taart-, gestapelde-, lijn-, balk-, en kolomgrafieken, zie afb. 1.



afb. 1 taartdiagram

De gegevens die de basis vormen voor de grafiek worden geïmporteerd uit één van de volgende GEOS applicaties, GeoWrite, GeoCalc, GeoFile of uit een NotePad file. Nadat deze gegevens in de respectievelijke programma's zijn gegenereerd worden ze ingelezen in GeoChart. De actuele afmetingen van een grafiek is afhankelijk van de gebruikte printer, zo'n 7 bij 9 cm. Hoe gaat het vervaardigen van een "chart" nu in z'n werk?

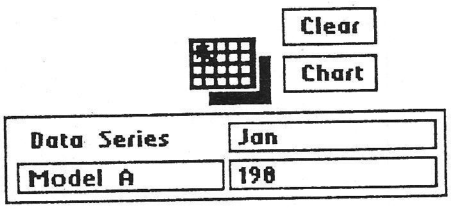


afb. 2 componenten

Het creëren van een grafiek

Eerst dient u de basisgegevens te vervaardigen. Deze kunnen bijvoorbeeld als resultaten uit een GeoCalc file afkomstig zijn. Indien u niet over deze krachtige spreadsheet beschikt, kunt u ook gebruik maken van de NotePad applicatie. Maar het meest voor de handliggend zijn natuurlijk gegevens afkomstig uit GeoCalc, omdat men doorgaans in een spreadsheet met getallen werkt. En deze getallen vormen de basis voor het GeoChart file. Wanneer u gegevens uit één van de eerder genoemde applicatie's gebruikt, dient u ze in een zgn. "textscrap" te copiëren.
Vervolgens wordt de "textscrap" ingevoegd in een GeoChart document. GeoChart verdeelt de gegevens in een viertal componenten, te weten de grafiektitel, de waarden, de serienamen en catagorienamen, zoals weergegeven in afb. 2.

Wanneer u eenmaal het onderscheidt weet te maken tussen de verschillende componenten, is eenvoudig te leren hoe deze componenten in een grafiek uit te zetten. Op het moment dat u gegevens invoegt zal de **data-modus** worden geactiveerd. Deze modus stelt u in staat de verschillende gegevens van de componenten te bekijken. Het **data-modus** dialoogvenster, zie afb. 3, geeft de verschillende waarden en de bijbehorende labels weer.



afb. 3 Data-modus

Men selecteert een gedeelte van de "textscrap" door een corresponderend hokje aan te klikken met het aanwijspijltje. Nadat u gereed bent met het bekijken van de gegevens, klikt u op het **chart**-icoon, waarna de grafiek zal worden getekend. In de **chart-modus** bestaat de mogelijkheid om de verschijningsvorm van de grafiek in verschillende opzichten te wijzigen. Ten eerste is er de **chart-style**. Met

deze keuze die uit de menubalk wordt geselecteerd bepaalt u of de grafiek moet verschijnen als area-, balk-, kolom-, taart-, lijn-, gespreid-punt-, gespreid-lijn-, punt- of stapelkolomgrafiek moet verschijnen. Met de optie **text** kunt u naar eigen believen verschillende labelnamen wijzigen. Het is hierbij niet alleen mogelijk om het letterfont, de stijl, grootte en inhoud te wijzigen, maar u kunt ook bepalen of de tekst al of niet moet worden weergegeven. Met de keuze **markers** kunt u de wijze waarop de waarden binnen de grafiek worden weergegeven aanpassen. Hierbij is het eveneens mogelijk de invulpatronen van de verschillende **markers** te veranderen. Er staan de gebruiker maar liefst 32 vulpatronen ter beschikking. Indien een **marker** bestaat uit een punt zijn slechts vier mogelijkheden aanwezig. **Axis values** (as-waarden) vormen automatisch de invoer voor GeoChart en zijn derhalve de basis voor de te maken grafiek. Tot de mogelijkheden behoort onder meer ook het invoeren van negatieve waarden. Daarnaast bestaat de mogelijkheid om de format van de waarden aan te passen, zodat deze als bv. dollars, procenten of decimalen worden getoond. Tenslotte kan de gebruiker het achtergrond-raster in drie verschillende verschijningsvormen wijzigen. Ten alle tijde kunnen de waarden in de **data-modus** worden geselecteerd.

Wat te doen met een grafiek?

Nadat een grafiek is ontworpen, conform de gebruikerswensen, bestaat de mogelijkheid deze te copiëren naar een ander document. GeoChart's file-management, die u in staat stelt een document te "updaten", te copiëren, van naam te veranderen of te printen, werkt op dezelfde manier als alle andere GEOS applicaties. Net zoals bij de andere GEOS applicaties, bestaat de mogelijkheid om de Desk Accessoires, zoals **calculator, Note Pad, clock** enz. te gebruiken, terwijl u met GeoChart bezig bent.

Conclusie

GeoChart is, evenals alle andere GEOS produkten, zeer prettig in het gebruik. Om daarvan een voorbeeld te geven, toen ik over een proefversie de beschikking had, kon ik zonder handleiding alle functie's van het programma uitvoeren, met een fraai resultaat. In aanvulling op GeoCalc is deze applicatie een must voor de serieuze GEOS gebruiker, die regelmatig met getallen werkt en resultaten op een goed ogende wijze wil presenteren. GeoChart en GeoCalc vormen tesamen een pakket, dat te vergelijken is met het PC-pakket **EXCEL**. GeoChart werkt op de Commodore 64/128, met dien verstande dat de 128 gebruikers eerst naar de 40-kolommodus moeten overschakelen. GeoChart wordt geleverd in een keurige doos, met één diskette en een duidelijke (helaas) Engelse handleiding, voor een prijs van fl. 89,00.

Bert Venema

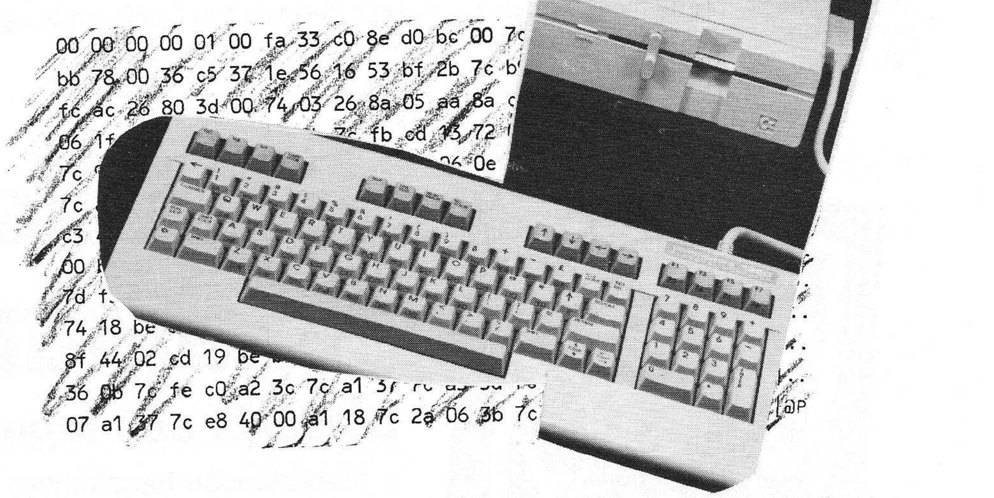
Zoals we het in het laatste nummer van het vorige jaar al vertelden, komen we nu in een stroomversnelling terecht. Dit keer gaan we het bijna magische gebeuren omtrent de BOOT sector nader bestuderen. Al vele keren viel het ons op dat er nog voldoende vraagtekens over de bootsector bestaan. Om aan al deze vragen een einde te maken zullen we eens proberen om de gehele boot zaak bloot te leggen.

Boot 128



Boot staat voor 'laars' en wordt gebruikt om bij MS-DOS computers het systeem in te laden. Het is als het ware een interface tussen systeem en hardware. Met de bootcall, wat zoiets betekend als aanroep, kunnen we een heleboel verschillende zaken ondernemen. Met deze bootcall kunnen we volautomatisch een programma inladen en opstarten. De gegevens over de in te laden software staat in de zogenaamde bootsector. De bootsector kunnen we bij de C128 vinden op track 1 sector 0. Om nu eerste de kreten track en sector, voor de nog niet ingewijde onder ons, te verklaren volgt de volgende uitleg.





Elke diskette heeft van deze tracks en sectoren. Een Commodore formaat diskette heeft 35 tracks. Elke track heeft sectoren, alleen is het aantal niet hetzelfde. Neem maar eens een diskette in de hand, en bekijk deze maar eens goed. Zo zult u twee inkepingen aan de onderkant van de diskette kunnen vinden. Deze zorgen er voor dat de diskettehoes precies op de juiste plaats blijft zitten in de drive. Het ovale gat aan de onderkant is de opening om de gegevens van schijf te kunnen inladen. Het grote gat in het midden is om de diskette door de drivemotor te laten rond draaien. Een diskette heeft dus ook een onder- en bovenkant! Dus heeft de diskette 35 sporen, zoals we de track ook wel noemen, aan de onder- en 35 sporen aan de bovenkant. In totaal dus 70. Een 1540/41 kan maar 35 sporen gebruiken omdat deze drive enkelzijdig is (Single Sided, SS). Met veel truukwerk kunnen ook 40 sporen op een diskette worden geplaatst.

Sectoren

De onderstaande tabel geeft het aantal sectoren per spoor aan:

```
spoor  1 - 17  21 sectoren.
spoor 18 - 24  19 sectoren.
spoor 25 - 30  18 sectoren.
spoor 31 - 35  17 sectoren.
```

De bovenstaande tabel laat dus direct de indeling van een diskette zien. Zo zijn er dus lange en korte sporen. Dus des te lager het spoor des te lager het aantal sectoren per spoor.

Directory

Wanneer we een nieuwe diskette pakken en deze formatteren met :"header"disknaam",i89 krijgen we een bruikbare diskette. Een nieuwe diskette moet dus altijd eerst worden geformatteerd. We brengen dus ons gebruikte formaat aan. De diskette wordt dus volgens het Commodore formaat ingedeeld. Na het formatteren zijn er 1328 sectoren, ook wel blocks vrij op de diskette. In totaal zijn er 1366 sectoren, maar we willen ook nog een inhoud opgave van de diskette hebben. Bij de dubbelzijdige diskettes,
(Double Sided, DS), zijn er omdat er ook twee kanten zijn, ook twee van deze inhoudsopgave nodig. Een op kant 0 en een op kant 1. Omdat bij het Commodore formaat de directory, zo wordt de inhoudsopgave ook wel genoemd, op spoor 18 staat gaan er per kant 19 sectoren af voor de directory. 19 omdat op spoor 18 19 van deze sectoren staan. Die 38 sectoren, voor de beide kanten 19, gaan van het totale vrije aantal sectoren af. 1366-38=1328 en zo komen we dan aan ons vrije aantal sectoren per dubbelzijdige diskette. De 38 sectoren worden gebruikt om filenamen in op te slaan. Ook staat daar onder andere het type van de file. Maar tot zover deze disk informatie.

Het booten

Bij het aanzetten van de C128 gebeuren in de computer een heleboel verschillende dingen. Een daarvan is de aanroep van PHOENIX. Dit is de aanroep, koude start, bij het aanschakelen van de C128. Deze routine, want dat is het gewoon, in de kernal van de computer begint op $FF56 Via dit adres gaat de routine naar $F867 En

```
1000 rem"*************************************************"
1001 rem"*   Boot-Block Writer voor de Commodore 128(d)   **"
1002 rem"*Door Johan & Johan voor COMMODORE INFO in 1989.  *"
1003 rem"*************************************************"
1004 :
1010 print chr$(14)chr$(147)+"BOOT-BLOCK WRITER"
1020 open 8,8,8,"#"
1030 open 15,8,15
1040 print#15,"b-a:"0,1,0
1050 input#15,a,b$,c,d
1060 if a=0 then goto 1120
1070 if a<>65 then print a,b$,c,d:goto 1220
1080 print "Block is in use. Overwrite (y/n) ?"
1090 get a$:ifa$=""then goto 1090
1100 if a$="y" then goto 1120
1110 goto 1220
1120 print#15,"b-p:"8;0
1130 :
1140 :for x= 0 to 255
1150 : a$=chr$(peek(4864+x))
1160 :  print#8,a$;
1170 : if st<>0 then goto 1220
1180 :next x
1190 :
1200 print#15,"b-p:"8;0
1210 print#15,"u2:"8;0;1;0
1220 close 8:close 15
1230 end
```

eenmaal hier aangekomen kontroleert de computer of er ook een module in de expansiepoort zit. Zo niet gaat hij verder met de BOOTCALL. Eerst worden het A- en X-register met een waarde geladen. In het X-register staat het device nummer van de drive. In de kernal staat er 8 voor het standaard device. Verder wordt er in het A-register een 0 ingeladen. De bootsector wordt in de bootbuffer, $0b00, via een block-read instructie ingeladen. Vervolgens wordt er gecontroleerd op de kenmerken CBM, wat staat voor Commodore Business Machines. De bootcall wordt afgebroken wanneer deze kenmerken niet worden aangetroffen.

De melding

U krijgt een melding van het systeem op uw scherm met de filenaam of de door de programmeur gewenste aanduiding. Deze melding wordt niet zoals Databecker verklaart in zijn INTERN, door BSOUT op uw scherm gezet maar door de kernal routine PRIMM! Waarschijnlijk was bij het uitgeven van de Intern nog niet alles bekend van de C128. U kunt dus zelf kiezen of u wel of niet een melding met filenaam wilt hebben. Hoe, daar komen we straks nog wel op terug. Het is dus theoretisch mogelijk om met alleen de melding 'Booting..." op het scherm een file in te laden. Om nog verder te gaan, en helemaal zonder melding op het scherm te komen, is de volgende truuk wel handig. U maakt van de filenaam de instructie scherm schoon! Dus $93 als filenaam opgeven, deze wordt dan door PRIMM om het scherm gezet en weg is de melding booting...

Het voorbeeld

Om onze uitleg wat duidelijker te laten overkomen hebben we het volgende bedacht. We laten het gewoon zien! Hieronder zult u zien wat er nu precies in die bootsector staat. Of liever gezegd wat er zou kunnen staan. U heeft wel een diskmonitor nodig!! Plaats op spoor 1 sector 0 de onderstaande drie bytes!

```
byte $00 $01 $02
      43  42  4d
```

Bij dit voorbeeld zal de bootcall eindigen in de monitor, met een break op $0b0b Maar de routine deed zijn werk en ging booten. Kijk nu maar een met de monitor vanaf $0b00 U zult zien dat de kenmerken vanaf dit adres te zien zijn. In de bootsector kunt u aangeven dat u een programma wilt inladen en opstarten. De volgende toevoeging zult u bij ons bovenstaande voorbeeld moeten aanbrengen om wat te kunnen inladen. Dus dit zijn de volgende vier bytes die na de drie hierboven genoemde bytes moeten worden geplaatst.

```
byte $03 $04 $05 $06
      00  13  00  01
```

Door deze aanvulling zal er vanaf adres $1300, de bytes 03 en 04 omgedraaid, 1 blok, byte 06, in bank 0, byte 05 worden ingeladen. Ook hier volgt een break in de monitor op adres $0b0b. Samengevat zijn dus de eerste drie bytes het boot kenmerk 'CBM'. De volgende twee voor het adres waar de in te laden routine, software, moet komen te staan. De vijfde byte is om aan te geven in welke bank, 0 of 1, de software moet worden ingeladen. Byte 6 geeft aan hoeveel blokken, sectoren, er moeten worden ingeladen. Na dit voorbeeld zouden we een melding op het scherm kunnen laten verschijnen bij het booten. Wat dacht u van:"Boot demo by J&J" Er moeten dus weer wat bytes achter de reeds door u ingevoerde bytes op spoor 1 sector 0 komen te staan. U plaatst de data dus vanaf $07 !!

```
byte $07 enz enz
       93 c2 4f 4f 54 44 45
4d 4f 20 42 59 20 ca 26 ca
00 00 00
```

Door deze toevoeging krijgt u ongeveer de melding op uw scherm:"Bootdemo by J&J...". En weer kwam u in de monitor, en om daar een einde aan te maken plaatsen we achter de melding een afsluiter '00' en daarachter de volgende data:

```
00 00 4c 03 40
```

Hierdoor springt na het booten de bootcall routine naar $4003 en krijgt u een cursor met daarboven 'ready'. Maar bij een koude start staat de karakterset op UPPERCASE en daarom kunt u het beste voor de 93 een 0e neerzetten. Dus ervoor en niet in plaats van !! Hierdoor wordt eerst de karakterset in de LOWERCASE gezet en daarna het scherm gewist. Na het wissen van het scherm komt dan onze melding op het scherm te staan. Zoals gezegd met PRIMM en niet met BSOUT en daarom dient u uw tekst met een 00 af te sluiten. De eerst volgende 00 geeft aan dat er geen file met een filenaam hoeft te worden ingeladen. Een heleboel disk beveiligingen doen zo hun eerste werk! Eerst laden ze een file in, die niet direct zichtbaar is en daarin staat dan ook heel vaak de software beveiliging! De bytes 4c 03 40 staan voor 'JMP$4003' waar de normale warme start begint. U zou in plaats van JMP$4003 dus ook naar een ander nuttig, of onnuttige, routine springen. De sector die door ons voorbeeld wordt ingeladen is sector 1 van spoor

```
1000 rem"**********************************************"
1001 rem"**       Boot-block-reader voor de C-128(d)     **"
1002 rem"**Door Johan & Johan voor COMMODORE INFO in 1989.**"
1003 rem"**********************************************"
1004 :
1010 color6,1:printchr$(147)chr$(14)chr$(5)
1020 print "BOOT-BLOCK READER"
1030 open 8,8,8,"#"
1040 open 15,8,15
1050 print#15,"u1:"8;0;1;0
1060 for x=0 to 255
1070 get#8,a$
1080 poke 4864+x,asc(a$+chr$(0))
1090 next x
1100 close 15:close8
1110 block=4864:a=0
1120 for x=0 to 2
1130 a=a+peek(block+x)
1140 next x
1150 if a<>210 then print "No boot-block":end
1160 if peek(block+3)=0 and peek (block+4)=0 then goto 1220
1170 a=(peek(block+3)+256*peek(block+4))
1180 print"Beginadres dec:";a
1190 print"Beginadres hex: ";hex$(a)
1200 print "Bank            :";:print(peek(block+5))
1210 print "Blocks          :";:print(peek(block+6))
1220 print "Name            :";
1230 byte=7:gosub 1350
1240 b=8+a
1250 if peek (block+b)=0 then print "No filename":goto 1280
1260 print "Filename        :";
1270 byte=b:gosub 1350
1280 b=b+a+1
1290 print
1300 a$=hex$(peek(block+b))
1310 print right$(a$,2);chr$(32);chr$(32);:b=b+1
1320 if b=255 then goto 1340
1330 goto 1300
1340 end
1350 a=0
1360 x=peek (block+byte+a):if x=0 then goto 1400
1370 if x<64 or (x>128 and x<161) then goto 1390
1380 print chr$(x);
1390 a=a+1:goto 1360
1400 print
1410 return
```

1. Wilt u twee sectoren inladen, door 02 op byte 6 neer te zetten, dan zal de volgende sector 2 van spoor 1 zijn. En zo gaat het als maar verder.

En verder

We kunnen dus zelf uitmaken in welke bank en vanaf welk adres onze routine wordt ingeladen. Ook kunnen we zelf beslissen waar de routine moet opstarten. Een goede programmeur kan dit zelfs zo goed wegwerken dat u niet eens kan zien waar zijn routine opstart. Hij zou dit bijvoorbeeld met illegale opcodes kunnen doen. En zeker voor de niet geheel ingewijde programmeur kan dit zeer frustrerend werken. We kunnen bijvoorbeeld gewoon even overschakelen naar de C64 mode, zonder dat de software gebruiker het in de gaten heeft. Een voorbeeld van een dergelijk truuk is CHESSMASTER 2000. Welke dus gewoon een C64 programma op 40 cls is!!

Virus

Nu weet u dus ook direct hoe een VIRUS te werk gaat en kunt u daar tegen in het harnas gaan. Met de besproken informatie zou u ook een VIRUS kunnen maken maar dat is niet ons doel. Bovendien weet iedereen nu ook hoe het werkt, en heeft het dus toch geen zin meer.

De boot-block reader

Met behulp van block read wordt vanaf track 1 sector 0 het bootblock ingeladen , deze wordt dan vervolgens in het geheugen op adres $1300 weggezet. Eerst wordt er gecontroleerd op CBM om te zien of er een bootblock op de schijf staat, is dit niet het geval dan zal dit worden aangegeven met NO BOOT BLOCK. Is dit wel het geval dan gaat het programma verder. Nu worden byte $03 en $04 binnen gehaald om te kijken of er een startadres wordt aangegeven om te zien waar het programma begint. Het start- en eindadres worden zowel decimaal als hexadecimaal aangegeven.

Nu wordt gekeken in welke bank het moet worden ingeladen en hoeveel blocks het programma lang is. Hierna wordt er naar een titel gezocht en indien aanwezig op het scherm gezet (bv. Booting Chessmaster 2000). Vervolgens wordt er naar een eventuele filenaam gezocht om het eventuele programma in te laden. En tenslotte wordt de bijbehorende hexadecimale code op het scherm afgeprint. U kunt dit ook allemaal weer terugvinden op adres $1300, waar het bootblock zich op dat moment bevindt.

Demo-bootblock

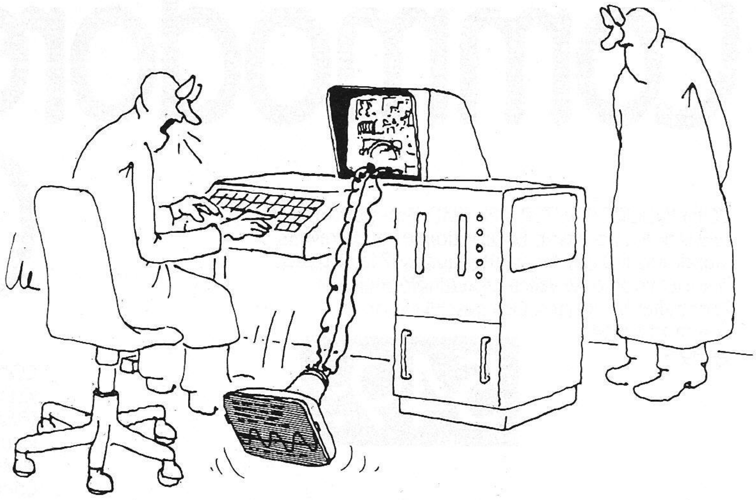
Het volgende programma is een demo. Deze demo laat zien dat je een programma (in dit geval een keuze menu) van uit de bootsector kunt opstarten. Nadat u de DEMO-BOOT-BLOCK heeft ingetypt doet u het volgende: U start het programma op , nu wordt dit bootblock via de data loader omgezet in machinetaal, die u op $1300 in het geheugen kunt vinden. Nu rest ons alleen nog het inladen van de boot-writer en het op te starten. Nu wordt het demo bootblock uit het geheugen van de 128 op schijf (track 1, sector 0) weggeschreven. Boot intikken en Hoppa daar is uw demo bootblock. U hebt nu de keuze uit een drietal mogelijkheden namelijk:

1. C-64
2. C-128 basic
3. C-128 monitor

Door de eerste optie te kiezen gaat u naar de 64 mode. Het programma springt nu namelijk naar de kernal routine GO64, die staat op adres $E24B. De tweede en derde optie spreken voor zich, in het eerste geval gaat u dus gewoon naar de basic en meldt de computer zich met ready. In het tweede geval springt u direct in de monitor. Het keuzemenuprogramma kan dus op elke schijf worden weggeschreven, maar denkt u er wel om dat er geen andere bootsector op de schijf aanwezig is. U bent dan namelijk alle informatie die in deze bootsector stond kwijt. Om dit te controleren kunt u dus de boot-block reader gebruiken.

Anti Reset 128

Dit programma zal u niet nieuw in de oren klinken. Nadat dit programma werd geplaatst hebben we vele vragen gehad of de anti reset niet in de bootsector kon worden geplaatst. Omdat we toch al bezig waren met de bootsector bloot te leggen hebben we dit dan maar even geregeld. Het programma gaat precies het zelfde te werk als bij de boot-menu writer. U kunt dus de anti reset 128 op elke schijf in de bootsector wegschrijven.



Boot-block writer

Zoals we al eerder in de tekst hadden opgemerkt hebben we een handige boot-block writer geschreven. dit programma staat elders in het blad afgedrukt. Met behulp van deze kleine utility en de informatie die wij u in dit artikel hebben gegeven kunt u zelf uw eigen boots maken. U beschouwd het geheugen deel vanaf adres \$1300 tot \$1400 als track 1 sector 0 van de diskette. Met een eenvoudige machinetaalmonitor maakt u dan het bootblock zoals u dat hebben wil in het geheugen van uw 128. Vervolgens kunt u deze dan weer met de boot-block writer wegschrijven. Ditzelfde wordt immers ook gedaan bij het demo-bootblock programma. Ook kunt u net als wij met de demo-boot block gedaan hebben deze doormiddel van een data-loader op \$1300 zetten, aan u dus de keus!

Tot slot

Wij hopen dat wij u met deze informatie en programmatuur van dienst zijn geweest, en dat al uw vragen over de bootsector beantwoord zijn. Als u nog handige tips heeft over de bootsector laat het dan even weten ja, dan kunnen de andere lezers er ook van mee profiteren.

```
1000 rem"**********************************
    ***** ************"
1010 rem"* ANTI RESET 128 in de bootsector
    voor de C128(d) *"
1020 rem"* door Johan & Johan voor COMMODORE
    INFO 1989.      *"
1030 rem"************************************
    *****************"
1040 :
1050 print chr$(14)chr$(147)+"ANTI RESET 128
    BOOT WRITER"
1060 :for x = 4864 to 5087
1070 : read a
1080 :  cs=cs+a:poke x,a
1090 : poke x,a
1100 :next x
1110 :
1120 if cs<>20832 then printchr$(7):list:end
1130 :
1140 open 15,8,15
1150 open 8,8,8,"#"
1160 print#15,"b-p:"8;0
1170 print#15,"b-a:"0;1;0
1180 input#15,a,b$,c,d
1190 if a=0 then goto 1260
1200 if a<>65 then print a,b$,c,d:goto 1340
1210 print "Block is in use. Overwrite (y/n)
    ?"
1220 get a$:if a$="" then goto 1220
1230 if a$="y" then goto 1260
1240 goto 1340
1250 :
1260 :for x=0 to 255
1270 : a$=chr$(peek(4864+x))
1280 :  print#8,a$;
1290 : if st<>0 then goto 1340
1300 :next x
1310 :
1320 print#15,"b-p:"8;0
1330 print#15,"u2"8;0;1;0
1340 close8:close15
1350 end
1360 :
1370 data 067,066,077,000,019,000,001,065,
    078, 084,073,045, 082,069,083,069
1380 data 084,032,049,050,048,000,000,162,
    210, 189,036,011, 157,255,018,202
1390 data 208,247,076,000,019,169,248,133,
    250, 169,255,133, 251,169,250,141
1400 data 185,002,169,044,162,001,160,000,
    032, 119,255,169, 249,133,250,169
1410 data 255,133,251,169,250,141,185,002,
    169, 019,162,001, 160, 000,032,119
1420 data 255,032,009,225,032,123,192,032,
    061, 246,201,239, 208, 003,032,139
1430 data 248,036,215,016,005,169,001,076,
    071, 019,169,000, 141, 048,208,162
1440 data 000,189,110,019,157,000,016,232,
    224, 076,208,245, 032, 125,255,014
1450 data 147,193,078,084,073,032,210,069,
    083, 069,084,032, 049, 050,056,000
1460 data 076,003,064,006,006,012,012,005,
    005, 006,004,004, 006, 147,070,065
1470 data 083,084,013,147,083,076,079,087,
    013, 068,076,079, 065, 068,009,009
1480 data 009,032,032,032,013,066,076,079,
    065, 068,009,009, 009, 032,032,032
1490 data 013,147,068,073,114,013,147,082,
    085, 078,013,147, 076, 073,083,084
1500 data 013,077,079,110,013,147,066,111,
    013, 147,072,069, 076, 080,013,255
```

Computer Aided Design is niet meer uit de moderne architectuur en industrie weg te denken. Van de bouwtekeningen voor uw zomerhuisje, electronische circuits voor audio-apparatuur, lenzenstelsels voor camera's tot spaceshuttles, alles rolt tegenwoordig uit de door CAD-software gestuurde computer. Vroeger was CAD een toepassing voor grote systemen. De laatste tijd is de CAD-techniek echter in een stroomversnelling geraakt en verschijnen steeds meer systemen voor de kleine kantoor- en microcomputergebruiker. Ook voor de Commodore C-128 (en natuurlijk de Amiga en CBM PC's) zijn inmiddels een aantal CAD-pakketten beschikbaar gekomen.

CAD op de C-128

Een kennismaking met Computer Aided Design



CAD ging van mainframe naar minicomputer en van minicomputer naar microcomputer. Een C-128 is uiteraard iets anders dan een PC. CAD op de Commodore kan daarom nooit zo uitgebreid zijn als op een groter systeem, maar als introductie tot de CAD-techniek is een Commodore-huiscomputer zo gek nog niet. Wij zien CAD op de Commodore C-64 en C-128 daarom als leerpakketten die de hobbyist en student de mogelijkheden van CAD laat zien. Bovendien zijn een aantal tekeningen op kleinere schaal, bijvoorbeeld de bouwplannen voor een schuurtje of een enkel mechanisch onderdeel, best bruikbaar. Huiskamer-CAD slaat in deze naast kantoor-CAD zeker geen slecht figuur.

Wat is CAD?

Achter het begrip Computer Aided Design schuilt veel meer dan alleen het maken van technische tekeningen. Na installatie gaat voor de gebruiker een geheel nieuwe wereld open van grafische ontwerpen, illustraties (art work, dia's, overheadsheets) stroomdiagrammen, bedradingsschema's, layout van formulieren, bouwtekeningen, spreadsheet en tekstverwerkings-toepassingen. Betrekkelijk nieuw zijn de Amiga bekende simulaties waarbij de klant bijvoorbeeld op de monitor kan bekijken hoe zijn toekomstige bungalow of fabriekje er uit komt te zien. De mogelijkheden hangen uiteraard van de hard- en softwareconfiguraties af, maar zelfs met een eenvoudig Commodore XT-tje of een C-128 en een goedkoop CAD-pakket kan nog menige serieuze gebruiker uit de voeten.
De popularising van de PC en huiscomputer heeft CAD in haar kielzog meegetrokken. Werd CAD in het verleden door specialisten op een mini- of mainframe-computer beoefend, tegenwoordig is er geen sprake meer van eiland-vorming, want iedereen kan zelfstandig met de eigen PC aan de slag.

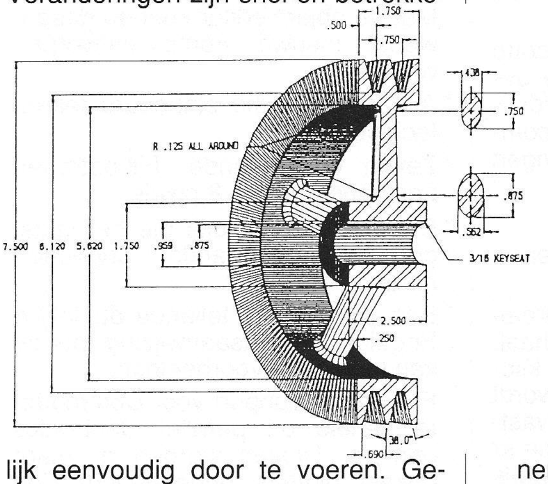
In eenvoudig Nederlands betekent CAD met de computer architectonische en industriële ontwerpen maken op het beeldscherm. Dat klinkt gemakkelijker dan het lijkt, want niet iedereen is zo maar een geschoold ontwerper. CAD-pakketten helpen de (on)ervaren gebruiker gelukkig op tal van manieren. Er is altijd sprake van een of ander coördinaten-systeem dat de juiste afmetingen en dimensies (automatisch) vastlegt. Een schaalverdeling helpt bij het bepalen van de juiste maten, zodat de tekening later qua schaal exact met de werkelijke afmetingen overeenkomt.

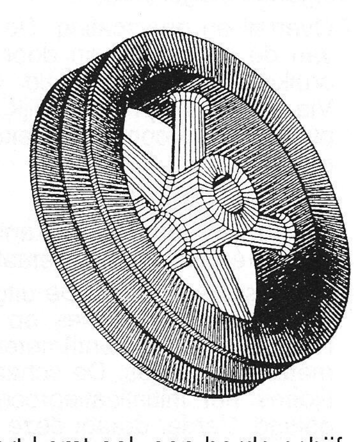
Wie twee linker tekenhanden heeft hoeft bij eenvoudige CAD-toepassingen niet veel te vrezen. De software tekent lijnen en standaardfiguren of vormen automatisch na het invoeren van de bijbehorende coördinaten. Lastige veelhoeken, circels, bochten, en loodrechte hoeken behoeven dus geen tekenprobleem meer te zijn. Hetzelfde geldt voor het aanbrengen van Fill- patronen, schaduweffecten en het simuleren van 3D.
Vele CAD-pakketten beschikken zelf over complete symbool- en figuurbibliotheken met huisje, boompje, beestje en vele standaard industriële vormen. Ook zijn er dikwijls verschil-

lende Fonts ingebouwd voor het op attractieve wijze verduidelijken van de verschillende onderdelen van het ontwerp.
Zoom en scrollcommando's maken het mogelijk om een veel groter ontwerp te maken dan op het monitorscherm past en details voor nauwkeurige bestudering en correctie uit te vergroten.
Cad programma's zijn objected-based. Dat wil zeggen de software behandelt de verschillende tekenobjecten niet als een collectie beeldpunten (pixels), maar als een vaste vorm die over het scherm verplaatst, vergroot/verkleind, geroteerd, gespiegeld, vervormd, of gedupliceerd kan worden.

De voordelen

Computer Aided Design biedt ten opzichte van de conventionele tekentafel vele voordelen. Een vergaande standarisatie verhoogt de nauwkeurigheid van de tekeningen en gebruikt steeds dezelfde tekensymbolen. Daarmee worden de individuele verschillen tussen de verschillende technische tekenaars verminderd en ontstaan ontwerptekeningen met een hoge consistentie en goede kwaliteit. Veranderingen zijn snel en betrekkelijk eenvoudig door te voeren. Gewoon de oude tekening inladen, de wijzigingen aanbrengen en een nieuw up to date ontwerp rolt uit de plotter. Kortom, CAD staat borg voor een snellere en efficiëntere bedrijfsvoering.
In de industrie wordt CAD veel gebruikt voor het electronische testen van het gemaakte ontwerp. Simulatie van stress en andere ontwerponvriendelijke factoren zoals hitte, druk, wind en golven maken een tijdige aanpassing van het ontwerp mogelijk. Constructiefouten komen nu zonder het in gevaar brengen van mensenlevens of kapitaalgoederen aan het licht. Een andere aantrekkelijke optie is het testen van reeds bestaande constructies op toekomstige, onvoorziene, belastingen. Bijvoorbeeld een brug die voor 10-tonners gemaakt werd en nu 20-tonners moet torsen. Men noemt deze testmethode wel Finite Element Analyses. In de ruimte worden een aantal draagcoördinaten opgetekend en daarna volgens het ontwerp tot een ruimtelijke structuur aangekleed. Nu worden met een analyseprogramma krachten op deze constructiepunten uitgeoefend en de (mogelijke) gevolgen daarvan doorgerekend.
Tot slot noemen wij nogmaals de grotere toegankelijkheid van CAD- systemen voor de niet technische tekenaar.





De basisconfiguratie

Een C-128 computer met bijbehorende software is alleen niet voldoende voor het bouwen van een CAD-configuratie. Daar komt heel wat meer bij kijken. In het algemeen bestaat een thuis- of school-CAD- systeem uit de volgende componenten:

- Een C-128 met floppy disk en voldoende vrij RAM. Voor gedetailleerde ontwerpen is een RAM-uitbreiding tot 256K wenselijk. Binnenkort komt ook een harde schijf voor de C-128 en C-64 beschikbaar en die kan heel wat CAD-data aan.
- Een goede HIRES monitor is vrijwel onmisbaar. Er is keuze uit monochroom- of kleurentypen. De laatste is aanzienlijk duurder, maar maakt het beeld door kleurschakeringen en - kontrasten er wel duidelijker en overzichtelijker op.
- Een CAD-pakket dat aan de eigen eisen voldoet. Niet iedereen gebruikt CAD voor hetzelfde doel en in vele gevallen gaat het zelfs om meerdere toepassingen. Net als elke andere software- aankoop is het van groot belang om een prioriteitenlijstje te maken. Aan toeters en bellen heeft de gemiddelde gebruiker niet zoveel. Wel aan een gebruiksvriendelijk pakket dat precies doet wat het moet doen en nog redelijk geprijsd is ook. Het hier besproken CADPAK-128 voldoet aan de meeste van deze eisen.
- Input-devices. Het tekenen met een CAD-pakket kan met tal van invoerapparatuur of "stuurwijzers". Voor serieus gebruik zijn de cursortoetsen van het keyboard eigenlijk totaal ongeschikt. Zelfs als tijdelijke noodoplossing. Vrijwel elk CAD- pakket werkt met een muis. Helaas geven deze digitale knagers nogal een last. Onbetrouwbare overbrenging, slechte compatibiliteit met hard- en software en het crashen van het systeem bij het per ongeluk uitvoeren van een verboden manouvre vormen slechts een kleine greep uit de vele gruwelen waarmee de beginnende CAD-man/vrouw geconfronteerd kan worden. Dat hoeft echter niet als men van meet af aan met een goede C-128 compatibele muis en stuursoftware start.

Behalve muizen is er wat de stuurwijzers betreft nog keuze uit lichtpennen, graphic tablets, joysticks en track/joy-ballen. Elk device heeft zo zijn eigen enthousiaste aanhangers/boe-roepers, voor- en nadelen, en mogelijk/onmogelijkheden. Het valt moeilijk om hierin verantwoord te adviseren. Dikwijls vindt u in het manual de nodige aanbevelingen en afraders die een verstandige keuze vergemakkelijken.

- Enkele CAD-pakketten ondersteunen scanners en digitizers. Daarmee kunnen reeds bestaande ontwerpen in het CAD- systeem ingevoerd en gewijzigd worden. Een nuttig tool voor de bouwkundig- of industrieel-tekenaar die nog een kast vol oude ontwerpen had liggen of het reklamebureau dat bestaande afbeeldingen in een nieuwe campagne wil integreren. Ook voor de C-128 en C-64 zijn met name in Duitsland diverse scanners en digitizers met tal van artistieke en CAD-mogelijkheden te koop.
- Voor het maken van perfecte hardcopies komen zowel plotters,

printers als beeldrecorders in aanmerking. Van oudsher is de plotter het geeigende tekeninstrument voor CAD/CAM- toepassingen. Een goede plotter kan een zeer hoge grafische resolutie met een perfecte afdrukkwaliteit bereiken. Voor een plotter geldt in feite hetzelfde als bij de monitor. Een inferieur model haalt niet uit het systeem wat er in zit. Behalve papieren hardcopies kunnen de meeste plotters ook transparante folies (overheadsheets) beschrijven. Voor de Commodore is er keuze uit de, eigenlijk niet serieus te nemen, op papierrol schrijvende 1520 en aanverwante plotters waarvan het papierformaat veel te klein is en de redelijk geprijsde Comx PL- 80.

Wie niet zo aan de optimale kwaliteit tilt en even snel een redelijk tot goed hardkopietje wil uitdraaien is bij een 24-naalds kleuren-matrixprinter of een goede inkjet-printer aan het juiste adres. Een alternatieve keuze is bijvoorbeeld de nieuwe Commodore MPS 2000 C kleurenprinter. De moderne printers zijn grafisch heel wat mans, maar kunnen nog niet echt aan de plotter tippen.

° Koppeling aan een database. Met behulp van een database kan een schat aan grafische en alfanumerieke gegevens aan het CAD-ontwerp gekoppeld worden. De dienst publieke werken van een gemeente kan zo bijvoorbeeld het rioleringssysteem in kaart brengen en bij alle buizen de diameter, diepte, gebruikte materialen, ouderdom, onderhoudsgevoeligheid, de bij onderhoud af te sluiten gedeeltes e.d. vermelden.

CADPAK-128

Het Britse softwarehuis Adamsoft heeft onlangs het HIRES ontwerp- en tekenpakket CADPAK-128 op de markt gebracht. Dit disk-based CAD-pakket tekent 40-koloms grafische beelden op de Commodore monitor en biedt printerdrivers voor 640 x 360 dots hardcopies.
De gebruiker heeft de keuze uit twee displayscreens:

° Het Main Screen van 640 x 360 pixels. De werkelijke resolutie in deze display-modus bedraagt 320 x 200 pixels, maar de gebruiker heeft gewoon een twee maal zo groot schermoppervlak tot zijn/haar beschikking.

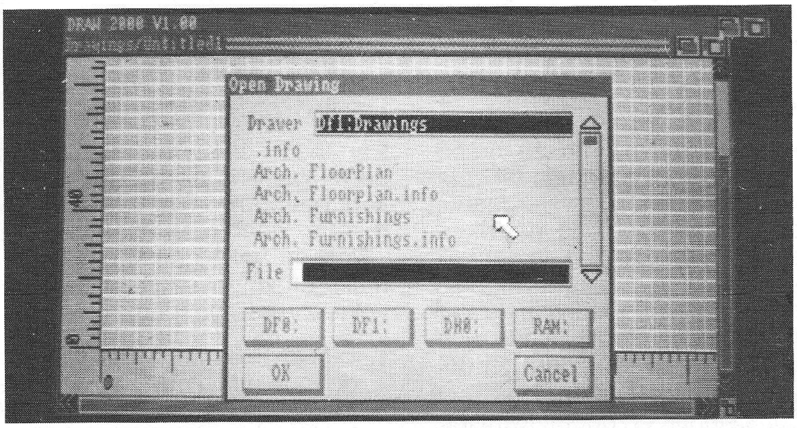
° Het gewone scherm van 320 x 200 pixels.

Via het screen window (beeldvenster) kunnen objecten vrijelijk tussen de beide displays gekopieerd worden.
Het tekenen zelf kan met de cursortoetsen, een muis of een lichtpen. De lightpen-optie is zeker de moeite van het overwegen waard daar het gebruikersgemak, snelheid en de mogelijkheden van de software er aanzienlijk mee vooruitgaan. Het aansluiten van de pen ging vrijwel probleemloos, gewoon een kwestie van inpluggen, evenals het instellen met behulp van het lightpen-menu.

° Templates zijn lijntekeningen die voor later gebruik gesaved kunnen worden. Zo kunt u de bibliotheek zelf verder uitbreiden. Het bijzondere van de templates is dat zij weer in andere tekeningen ingebouwd kunnen worden. Templates zijn van uit BASIC aan te roepen en zo ook in zelfgemaakte programma's in te bouwen.

° dupliceervenster. Via het schermvenster (window) kan de ontwerper een object kopiëren, roteren of spiegelen en daarbij ook nog in grootte wijzigen.

° Drie Fonts met tekst in vier verschillende grootten en 3D- letters. Met de object editor kunnen desgewenst nieuwe Fonts ontworpen worden.



CADPAK-128 biedt de gebruiker de volgende mogelijkeden:

° Overlay en printscaling. De grootte van de hardcopy kan door de gebruiker vrijelijk ingesteld worden. Via overlays zijn behoorlijk gecompliceerde constructietekeningen mogelijk.

° Volledige menusturing.

° Saven van gemaakte tekeningen in BASIC en 6502 machinetaal.

° Exact scaled output. De uitgedraaide tekening is precies op schaal. Naar keuze in (centi)meters, kilometers en inches. De schaal wordt tijdens het initialisatieproces vastgelegd. Verder draagt deze optie er zorg voor dat cirkels ook als cirkels en niet als eieren afgedrukt worden.

° Het object management systeem (OMS) hanteert beelden van 16 x 16 pixels op het normale scherm. De gebruiker kan deze objects voor latere toepassingen saven. Er kunnen maximaal 104 OMS- objects tegelijk gebruikt worden. Onder het OMS vallen ook enkele standaardbibliotheken op de systeemschijf met onder andere kaarten, meetkundige figuren en electronische symbolen.

° Alle gebruikelijke standaard tekentechnieken.

° Zeven verschillende -Fill-patronen bestaande uit 8 x 8 pixels.

° De Try Again-functie die het laatst gegeven commando ongedaan maakt.

° Een 86-pagina's tellende duidelijke Engelse gebruiksaanwijzing met tal van lessen en voorbeelden.

° Full printer support voor Commodore seriële en parallel Centronics printers. Helaas worden er maar enkele plotters ondersteund! Een van de weinige nadelen van dit CAD-pakket.

CADPAK-128 kost rond de *f* 125,-. Daar komt de prijs van de lichtpen nog bij. Info bij: Adamsoft Ltd., 18 Norwich Avenue, Rochdale, Lancs. OL11 5JZ.
CAD op de C-128 is nu voor iedereen bereikbaar. Verwacht echter geen professionele kwaliteit, maar wel bruikbare en vooral leerzame resultaten. En om de prijs hoeft u het echt niet te laten.

PRINT OUT C-128 met o.a. Patience

Taxi

Bij een taxi centrale komen aan de lopende band meldingen binnen van klanten die op een taxi wachten. Het is de taak van de coordinator om lege taxi;s naar die plaatsen te sturen waar de klanten staan. De taxichaffeur op zijn beurt meldt aan de centrale waar de klant naar toe wil, en geeft door als de bestemming is bereikt.

Dit is in het kort wat er gebeurt bij het programma taxi. De coordinator zit uiteraard achter het toetsenbord van de commodore 128.

Het gebied dat wordt bestreken door de centrale bestaat uit tien plaatsen, A t/m J, en een garage aangegeven door een *. Er zijn 5 taxi's, genummerd van 1 t/m 5, en hebben elk een eigen kleur. Als er een melding van een klant binnen komt moet er zo snel mogelijk een taxi worden heengestuurd. Dit gebeurt door het nummer van de gewenste taxi in te geven, gevolgd door het nummer van de plaats waar hij heen moet. Als de taxi leeg is gaat hij op weg naar het op gegeven punt. Wordt er een opdracht gegeven aan een taxi die met een klant onderweg is, dan gaat deze op weg zodra hij zijn vracht heeft afgeleverd. Staat er op dat punt echter een nieuwe klant te wachten dan gaat het gegeven commando niet door.

Het scherm bestaat uit de volgende onderdelen: Een plattegrond van het gebied. Hier zijn de tien punten aangegeven en de positie van elke taxi.
Linksboven staat het totaal aantal wachtende klanten, daaronder per plaats hoeveel klanten er staan te wachten. Ook wordt hier aangegeven of er taxi's onderweg zijn naar dat punt of niet.
Linksonder staat informatie over elke taxi. Een enkele letter (of *) geeft aan dat de taxi op die plaats staat te wachten. Een letter voorafgegaan door "-" wil zeggen dat de taxi leeg onderweg is naar dat punt. Wordt de letter vooraf gegaan door "+" dan vervoert de taxi een klant. Daaronder staat tussen haakjes een vervolgopdracht. Middenonder staat het nummer van de taxi waarvoor het commande geldt. Rechtsonder staat de score en de resterende tijd.
Door het intoetsen van een P wordt het spel onderbroken, na het aanraken van een willekeurige toets gaat het spel weer verder. Een ? geeft het helpscherm, met een uitleg wat er zoal op het scherm gebeurt.

Punten worden verkregen door klanten te vervoeren. Elke taxi die leeg is, (rijdend of stilstaand), kost punten, behalve als de auto in de garage staat. Een wachtende klant kost ook punten, hoe meer klanten er staan te wachten des te meer strafpunten kost dat per klant.

Er is een keus uit vier verschillende druktes, rustig (2 taxi's), matig druk (3 taxi's), spitsuur (5 taxi's) en wisselend druk zodat er steeds een wisselend aantal taxi's moet worden ingezet.

Het spel moet binnen een bepaald tijdsbestek zijn gespeeld, de punten die hiervoor nodig zijn staan op het menuscherm.

De maker van dit uitgebreide programma is een regelmatige inzender voor deze rubriek de Heer Y. Rozijn uit Amsterdam.

```
1     rem          taxi
2     rem  voor c128 met 40koloms
3     rem  kleurenscherm
4     rem  door y.rozijn,amsterdam
5     rem
10    gosub1260:gosub850:gosub1080:do:go
      sub710:loop
20    rem **** opdracht ****
30    p=pand15:ift(t)thenchar1,1,2*t(t),
      "[SPACE]"
40    dp=x(p)-p(t):dq=y(p)-q(t):d=int(sq
      r(dp*dp+dq*dq)/4):a$=b$(b)+c$(p)
50    ifdthendp(t)=dp/d:dq(t)=dq/d:n(t)=
      d:t(t)=p:b(t)=b:char1,4*t-4,23,a$
60    return
70    rem **** kommando ****
80    char1,4*t-4,24,"("+c$(p)+")":v(t)=
      p+16
90    return
100   rem **** input ****
110   geta$:ifa$="p"thenchar1,21,24,a$:g
      etkeya$:char1,21,24,"[SPACE]":at=0
      :return
120   ifa$="?"thengosub310:return
130   i=instr("12345",a$):ifithenat=i:ch
      ar1,21,24,a$:else begin
140   i=instr("*abcdefghij",a$):ifi>0and
      at>0thenp=i-1:t=at:gosub80
150   bend
160   return
170   rem **** taxi pikt klant op ****
180   do:p=fnr(10)+1:loopwhilep=i:b=1:go
      sub30
190   pw(i)=pw(i)-1:char1,2,2*i,left$("W
      WWWWWW",pw(i))+"[SPACE]"
200   ifpw(i)=0thencolor1,16:char1,xc(i)
      ,yc(i),c$(i):color1,8
210   nk=nk-1:v(t)=0:char1,4*t-4,24,"[3x
      SPACE]":char1,0,0,right$(str$(nk),
      2)
220   return
230   rem **** nieuwe klant ****
240   i=fnr(10)+1:ifpw(i)<7thenbegin:pw(
      i)=pw(i)+1:nk=nk+1
250   char1,0,0,right$(str$(nk),2):char1
      ,2,2*i,left$("WWWWWWW",pw(i))
260   color1,2:char1,xc(i),yc(i),c$(i):c
      olor1,8:fk=fk+f
270   iffvandtd<txthenff=ff+1+4*(ff=4):f
      =f(ff):tx=td-150-fnr(100):fk=f:f=f
      /10
280   bend
290   return
300   rem **** help ****
310   h=0:do:fort=1to5:spritet,h:next:gr
      aphich:getkeya$:h=1-h
320   loopuntila$=chr$(13):fort=1to5:spr
      itet,1:next:graphic1
330   return
340   rem **** loop ****
350   sc=0:nk=0:fk=f:f=f/10:gosub240
```

```
360   do:dowhilepeek(208):gosub110:loop
370   : fort=1to5:i=t(t)
380   : : ifn(t)thenbegin:n(t)=n(t)-1
390   : : : ifn(t)thenbegin:p(t)=p(t)+dp
      (t):q(t)=q(t)+dq(t):a$="+"
400   : : : bend:elsebegin
410   : : : : p(t)=x(i):q(t)=y(i):a$="[S
      PACE]":char1,4*t-4,23,"[SPACE]":b(
      t)=0
420   : : : bend
430   : : : movesprt,p(t),q(t):ifithench
      ar1,1,2*i,a$
440   : : bend
450   : : ifn(t)=0andpw(i)>0thengosub180
460   : : ifb(t)thens=s+3:elsebegin:ifn(
      t)>0ori>0thens=s-1
470   : : : ifv(t)thenp=v(t):b=0:gosub30
      :v(t)=0:char1,4*t-4,24,"[3xSPACE]"
480   : : bend
490   : next:td=td-1:fk=fk-.1:ifrnd(1)*f
      k<1thengosub230
500   : s=s-nk*(.5+nk/nx):sc=sc+s:a$=mid
      $(a0$,sgn(s)+2,1):s=0
510   : char1,31,22,right$("[4xSPACE]"+s
      tr$(td),6)
520   : char1,31,24,right$("[4xSPACE]"+s
      tr$(int(sc)),6)+"[SPACE]"+a$
530   loopwhileabs(sc)<s0andtd>0
540   fort=1to5:spritet,0:next:color1,4:
      char1,20,8,"spel[SPACE]over"
550   i=3-2*(sc>0)-2*sgn(sc)*(abs(sc)>=s
      0):color1,8
560   forj=0to1:gshapez$(i+j),200-z(i+j)
      /2,100+12*j:next
570   do:getkeya$:loopuntila$=chr$(13)
580   return
590   rem **** scherm init ****
600   color1,16:graphic1,1
610   fori=0to10:char1,xc(i),yc(i),c$(i)
      :char1,0,2*i,c$(i):pw(i)=0:next
620   fort=1to5:color1,rsprite(t,1):gsha
      peta$,32*t-24,176:color1,8
630   char1,t*4-4,22,chr$(48+t):char1,t*
      4-3,23,"*":movesprt,x(0),y(0):spri
      tet,1
640   p(t)=x(0):q(t)=y(0):t(t)=0:n(t)=0:
      b(t)=0:v(t)=0:next
650   fori=1to8:p=7*i+14+(i=1):gshapew$(
      i),p,0:next
660   draw1,0,172to319,172:draw1,77,0to7
      7,172:draw1,152,172to152,199
670   draw1,190,172to190,199:draw1,152,1
      84to190,184:gshapetx$,156,174,2
680   char 1,25,22,"tijd:":char 1,25,24,
      "score:"
690   return
700   rem **** hoofdloop ****
710   x=1:y=1:color1,8:graphic1,1
720   fori=1to320step44:gshapeta$,i,0:gs
      hapeta$,i,172:gshapetx$,i+11,0
730   gshapetx$,i+11,172:next:color1,16:
      fori=0to39:char1,i,16,"[COM-D]":ne
      xt
740   color1,14:char1,9,3,"kies[SPACE]be
      zettingsgraad:":draw1,71,32to237,3
      2
750   char1,5,24,"?[SPACE]voor[SPACE]hel
      pscherm[SPACE]tijdens[SPACE]spel":
      poke 208,0
760   char1,5,19,"druk[SPACE]op[SPACE]re
      turn[SPACE]om[SPACE]te[SPACE]begin
      nen":color1,11:box1,36,149,275,162
770   color 1,16:char 1,16,14,"te[SPACE]
      behalen[SPACE]score[SPACE]~"
780   do:fori=1to4:color1,16+14*(x=i):ch
      ar1,7,2*i+4,c$(i)+"[2xSPACE]"+f$(i
      )
790   char 1,30,2*i+4,right$(str$(s0(i))
      ,4):next
800   getkeya$:i=instr("abcd",a$):ifithe
      nx=i
810   loopuntila$=chr$(13):fv=(x=4):iffv
      thenf=f(1):tx=1850-fnr(100):elsef=
      f(x)
820   td=2000:s0=s0(x):nx=nx(x):ff=1:gos
      ub 600:gosub 350
830   return
840   rem **** init ****
850   color0,1:color4,1:color1,1:with2:g
      raphic1,1
860   fori=8192to8206:readx:pokei,x:next
870   shapeta$,0,0,23,20:fort=1to5:sprsa
      veta$,t:next:shapeta$,0,0,10,7
880   deffnr(i)=int(rnd(1)*i):x=rnd(-ti)
890   scnclr:fori=8192to8222:readx:pokei
      ,x:next:shapetx$,0,0,31,7
900   scnclr:char1,0,0,"wachtend":fori=1
      to8:shapew$(i),8*i-7,0,8*i,7:next
910   fori=0to10:readxc(i),yc(i):x(i)=8*
      xc(i)+23:y(i)=8*yc(i)+50:next
920   fort=1to5:readk:spritet,0,k,1,0,0,
      0:next:a0$="<[SPACE]>"
930   b$(0)="-":b$(1)="+":fori=0to10:c$(
      i)=mid$("*abcdefghij",i+1,1):next
940   fori=1to8:reada$:z(i)=8*len(a$):ch
      ar1,0,0,a$:shapez$(i),0,0,z(i)-1,7
      :next
950   fori=1to4:readf(i):next:fori=1to4:
      readf$(i):next
960   fori=1to4:reads0(i):next:fori=1to4
      :readnx(i):next
970   return
980   data30,127,97,255,222,255,97,0,0,1
      28,128,192,192,192,128
990   data0,255,219,24,24,24,61,0,0,24,6
      0,102,255,195,231,0,0,57,25
1000  data15,15,25,185,0,0,222,140,12,12
      ,140,222
1010  data10,,12,5,15,19,30,,21,17,39,9,
      31,14,20,10,22,2,37,6,32,20,8,11,4
      ,14,13
1020  dataonder jouw leiding zou,de zaak
       snel failliet zijn!,jammer!
1030  dataje hebt het niet gered,je hebt
       de zaak maar net,draaiende kunnen
       houden
1040  dataje staat aan het hoofd van,een
       goed lopend taxibedrijf
1050  data30,18,10,18,rustig,"matig[SPAC
      E]druk",spitsuur,"wisselend[SPACE]
      druk"
1060  data 500,1000,2000,1000,7,5,4,5
1070  rem **** helpscherm opbouwen ****
1080  scnclr0:printchr$(142)"[CTRL-8]..[
      SPACE][CTRL-4]aantal[CTRL-8]Y[COM-
      8]*[SPACE][CTRL-4]is[SPACE]de[SPAC
      E]garage":print"[CTRL-4]wachtenden
      "
1090  print"[CTRL-8][9xSPACE]Y[COM-8]a-j
      [SPACE][CTRL-4]zijn[SPACE]de[SPACE
      ]10[SPACE]plaatsen":print"[COM-8]a
      [CTRL-8]+WWW[4xSPACE]Y"
```

```
1100  print"[CTRL-8][9xSPACE]Y[CTRL-4][S
      PACE]kommando[SPACE]geven:":print"
      [COM-8]a[SPACE][CTRL-4]=plaats[CTR
      L 8]Y[CTRL-4][SPACE]Q[SPACE]selekt
      eer[SPACE]taxi";
1110  print"[SPACE](1-5)":print"[CTRL-8]
      [9xSPACE]Y[CTRL-4][SPACE]Q[SPACE]g
      eef[SPACE]plaats[SPACE](a-j[SPACE]
      of[SPACE]*)[SPACE]waar"
1120  print"[CTRL-8][SPACE]+[SPACE][CTRL
       4]=1[SPACE]of[SPACE][CTRL-8]Y[CTR
      L 4][3xSPACE]taxi[SPACE]naar[SPACE
      ]toe[SPACE]moet":print"[CTRL-4][3x
      SPACE]meer[2xSPACE][CTRL-8]Y[CTRL-
      4][SPACE]het[SPACE]komman";
1130  print"do[SPACE]komt[SPACE]tussen[S
      PACE]()":print"[CTRL-4][3xSPACE]ta
      xi's[CTRL-8]Y[CTRL-4][SPACE]linkso
      nderaan[SPACE]en[SPACE]gaat[SPACE]in"
1140  print"[CTRL-4][3xSPACE]op[SPACE]we
      g[CTRL-8]Y[CTRL-4][SPACE]zodra[SPA
      CE]de[SPACE]taxi[SPACE]leeg[SPACE]
      is.":print"[CTRL-8][9xSPACE]Y"
1150  print"[CTRL-8][2xSPACE]WWW[SPACE][
      CTRL-4]=[2xSPACE][CTRL-8]Y[CTRL-4]
      [SPACE]verdere[SPACE]toetsen:":pri
      nt"[CTRL-4][3xSPACE]wach-[SPACE][C
      TRL 8]Y[CTRL-4][SPACE]p[SPACE]=pau
      zeren;";
1160  print"[2xSPACE]?=helpscherm":print
      "[CTRL-4][3xSPACE]tende[SPACE][CTR
      L 8]Y[SPACE][CTRL-2][29xCOM-`]"
1170  print"[CTRL-4][3xSPACE]klan-[SPACE
      ][CTRL-8]Y[SPACE][CTRL-2][CTRL-9][
      SPACE]return[6xSPACE]:[SPACE]herva
      t[SPACE]spel[3xSPACE]":print"[CTRL
       4][3xSPACE]ten[3xSPACE][CTRL-8]Y";
1180  print"[SPACE][CTRL-2][CTRL-9][SPAC
      E]andere[SPACE]toets:[SPACE]wissel
      [SPACE]scherm[SPACE]":print"[CTRL-
      4][9xSPACE][CTRL-8]Y"
1190  print"[CTRL-8][9xSPACE]Y[CTRL-4][S
      PACE]nummer[SPACE]van[SPACE]de[SPA
      CE]taxi[SPACE]waarvoor":print"[CTR
      L 8][9xSPACE]Y";
1200  print"[CTRL-4][SPACE]kommando[SPAC
      E]geldt;[2xSPACE]p[SPACE]=pauze":p
      rint"[CTRL-8][9xSPACE]Y[11xSPACE][
      CTRL-4]~"
1210  print"[CTRL-8][21xSHIFT-*][CTRL-4]
      [SHIFT-—][CTRL-8][18xSHIFT-*]"
1220  print"[CTRL-8]+[SPACE][CTRL-4]:rij
      dt[SPACE]met[SPACE]klant[SPACE][CT
      RL 8][COM-G][SPACE][CTRL-4][SHIFT-
      -][CTRL-8][SPACE][COM-M]":print"[C
      TRL 8]-[SPACE][CTRL-4]:rijdt[SPACE
      ]leeg[6xSPACE][CTRL-8][COM-G][3xSP
      ACE][COM-M]"
1230  print"[CTRL-4][2xSPACE]:staat[SPAC
      E]stil[6xSPACE][CTRL-8][COM-G][SPA
      CE]1[SPACE][COM-N]";
1240  return
1250  rem **** inleiding ****
1260  color0,1:color4,1:color5,16:scnclr
      :printchr$(14)chr$(11)"Je[SPACE]be
      nt[SPACE]koor";
1270  print"dinator[SPACE]in[SPACE]een[S
      PACE]taxicentrale.Jouw[SPACE]taak[
      SPACE]is[SPACE]om[SPACE]lege[SPACE
      ]taxi's[SPACE]te[SPACE]dirigeren"
1280  print"naar[SPACE]die[SPACE]plaatse
      n[SPACE]waar[SPACE]klanten[SPACE]s
      taan[SPACE]te[SPACE]wachten.":print
1290  print"Er[SPACE]zijn[SPACE]10[SPACE
      ]plaatsen[SPACE](A-J),[SPACE]en[SP
      ACE]een[SPACE]garage(*).[SPACE]Er[
      SPACE]zijn[SPACE]5[SPACE]taxi's,[S
      PACE]";
1300  print"maar[SPACE]minder[6xSPACE]wa
      gens[SPACE]inzetten[SPACE]kan[SPAC
      E]ook.":print
1310  print"Een[SPACE]taxi[SPACE]levert[
      SPACE]alleen[SPACE]punten[SPACE]op
      [SPACE]als[SPACE]hijeen[SPACE]pass
      agier[SPACE]vervoert.[SPACE]";
1320  print"Een[SPACE]lege[SPACE]taxi[3x
      SPACE]kost[SPACE]punten,[SPACE]elk
      e[SPACE]wachtende[SPACE]passagier[
      3xSPACE]ook.[SPACE]Een[SPACE]";
1330  print"taxi[SPACE]in[SPACE]de[SPACE
      ]garage[SPACE]kost[SPACE]niets.":p
      rint:print"Een[SPACE]lege[SPACE]ta
      xi[SPACE]kun[SPACE]je[SPACE]";
1340  print"naar[SPACE]een[SPACE]ander[S
      PACE]puntsturen[SPACE]door[SPACE]h
      et[SPACE]nummer[SPACE]van[SPACE]de
      [SPACE]taxi[SPACE]in[SPACE]te"
1350  print"tikken,[SPACE]gevolgd[SPACE]
      door[SPACE]de[SPACE]letter[SPACE]v
      an[SPACE]het[2xSPACE]punt.[SPACE]A
      ls[SPACE]je[SPACE]een[SPACE]komman
      do[SPACE]";
1360  print"geeft[SPACE]aan[SPACE]een[SP
      ACE]taxi[SPACE]die[SPACE]een[SPACE
      ]vracht[SPACE]vervoert,[SPACE]word
      t[SPACE]het[SPACE]kommando[SPACE]";
1370  print"bewaard.[SPACE]Het[SPACE]gaa
      t[SPACE]dan[SPACE]in[SPACE]zodra[S
      PACE]de[SPACE]taxi[SPACE]leeg[SPAC
      E]is,[SPACE]tenzij[SPACE]op[SPACE]
      de[SPACE]";
1380  print"bestemmingvan[SPACE]de[SPACE
      ]taxi[SPACE]een[SPACE]nieuwe[SPACE
      ]passagier[SPACE]staat[2xSPACE]te[
      SPACE]wachten."
1390  print:print"[SPACE][CTRL-9][CTRL-8
      ][2xSPACE]druk[SPACE]op[SPACE]een[
      SPACE]toets[SPACE]voor[SPACE]het[S
      PACE]vervolg[2xSPACE][CTRL-0]";:ge
      tkeya$
1400  scnclr:color5,16:print"Het[SPACE]s
      cherm[SPACE]bevat[SPACE]de[SPACE]v
      olgende[SPACE]onderdelen:":print
1410  print"-[SPACE]Een[SPACE]plattegron
      d[SPACE]van[SPACE]alle[SPACE]punte
      n.[SPACE]Hier[3xSPACE]zijn[SPACE]o
      ok[SPACE]de[SPACE]posities[SPACE]v
      an[SPACE]";
1420  print"elke[SPACE]taxi[6xSPACE]aang
      egeven.":print:print"-[SPACE]Links
      [SPACE]staat[SPACE]hoeveel[SPACE]";
1430  print"passagiers[SPACE]op[SPACE]el
      k[3xSPACE]punt[SPACE]staan.":print
1440  print"-[SPACE]Linksonder[SPACE]sta
      at[SPACE]van[SPACE]elke[SPACE]taxi
      [SPACE]de[7xSPACE]plaats[SPACE]waa
      r[SPACE]hij[SPACE]is[SPACE]of[SPACE]";
1450  print"naar[SPACE]toe[SPACE]gaat,[4
      xSPACE]en[SPACE]of[SPACE]hij[SPACE
      ]leeg[SPACE]is[SPACE]of[SPACE]niet
      .":print
1460  print"-[SPACE]Middenonder[SPACE]st
```

```
        aat[SPACE]het[SPACE]nummer[SPACE]v
        an[SPACE]de[5xSPACE]taxi[SPACE]waa
        rvoor[SPACE]het[SPACE]komman";
1470    print"do[SPACE]geldt.[7xSPACE]Een[
        SPACE]'P'[SPACE]betekent[SPACE]pau
        ze.":print:print"-[SPACE]Rechtsond
        er[SPACE]";
1480    print"staan[SPACE]de[SPACE]restere
        nde[SPACE]tijd[4xSPACE]en[SPACE]de
        [SPACE]score.":print
1490    print"'P'[SPACE]toets:[SPACE]spel[
        SPACE]onderbreken.":print"'?'[SPAC
        E]toets:[SPACE]toon[SPACE]het[SPAC
        E]helpscherm."
1500    print:print"[SPACE][CTRL-9][CTRL-8
        ][3xSPACE]druk[SPACE]op[SPACE]een[
        SPACE]toets[SPACE]om[SPACE]te[SPAC
        E]beginnen[3xSPACE][CTRL-0]";:getk
        eya$
1510    return
```

Einde listing taxi

Checksum Taxi

REGEL 1	197	REGEL 530	228	REGEL 1100	136
REGEL 2	210	REGEL 540	11	REGEL 1110	142
REGEL 3	103	REGEL 550	217	REGEL 1120	167
REGEL 4	240	REGEL 560	87	REGEL 1130	106
REGEL 5	143	REGEL 570	217	REGEL 1140	200
REGEL 10	244	REGEL 580	142	REGEL 1150	228
REGEL 20	52	REGEL 590	213	REGEL 1160	122
REGEL 30	173	REGEL 600	81	REGEL 1170	196
REGEL 40	109	REGEL 610	44	REGEL 1180	50
REGEL 50	171	REGEL 620	221	REGEL 1190	165
REGEL 60	142	REGEL 630	207	REGEL 1200	26
REGEL 70	53	REGEL 640	54	REGEL 1210	213
REGEL 80	139	REGEL 650	237	REGEL 1220	159
REGEL 90	142	REGEL 660	66	REGEL 1230	200
REGEL 100	111	REGEL 670	110	REGEL 1240	142
REGEL 110	110	REGEL 680	159	REGEL 1250	114
REGEL 120	181	REGEL 690	142	REGEL 1260	237
REGEL 130	91	REGEL 700	137	REGEL 1270	44
REGEL 140	188	REGEL 710	13	REGEL 1280	124
REGEL 150	23	REGEL 720	121	REGEL 1290	118
REGEL 160	142	REGEL 730	188	REGEL 1300	183
REGEL 170	102	REGEL 740	50	REGEL 1310	127
REGEL 180	162	REGEL 750	240	REGEL 1320	204
REGEL 190	209	REGEL 760	194	REGEL 1330	25
REGEL 200	241	REGEL 770	236	REGEL 1340	38
REGEL 210	184	REGEL 780	150	REGEL 1350	247
REGEL 220	142	REGEL 790	101	REGEL 1360	67
REGEL 230	38	REGEL 800	64	REGEL 1370	68
REGEL 240	110	REGEL 810	58	REGEL 1380	254
REGEL 250	238	REGEL 820	237	REGEL 1390	238
REGEL 260	101	REGEL 830	142	REGEL 1400	71
REGEL 270	249	REGEL 840	19	REGEL 1410	150
REGEL 280	23	REGEL 850	1	REGEL 1420	11
REGEL 290	142	REGEL 860	55	REGEL 1430	9
REGEL 300	8	REGEL 870	1	REGEL 1440	252
REGEL 310	206	REGEL 880	169	REGEL 1450	178
REGEL 320	59	REGEL 890	240	REGEL 1460	217
REGEL 330	142	REGEL 900	174	REGEL 1470	156
REGEL 340	25	REGEL 910	211	REGEL 1480	146
REGEL 350	211	REGEL 920	111	REGEL 1490	252
REGEL 360	57	REGEL 930	206	REGEL 1500	29
REGEL 370	249	REGEL 940	137	REGEL 1510	14
REGEL 380	93	REGEL 950	34		
REGEL 390	30	REGEL 960	155		
REGEL 400	234	REGEL 970	142		
REGEL 410	121	REGEL 980	155		
REGEL 420	197	REGEL 990	95		
REGEL 430	80	REGEL 1000	180		
REGEL 440	139	REGEL 1010	183		
REGEL 450	114	REGEL 1020	247		
REGEL 460	53	REGEL 1030	126		
REGEL 470	3	REGEL 1040	166		
REGEL 480	139	REGEL 1050	206		
REGEL 490	184	REGEL 1060	101		
REGEL 500	14	REGEL 1070	57		
REGEL 510	174	REGEL 1080	102		
REGEL 520	177	REGEL 1090	221		

Patience

Kaart spelen hebben we al verschillende keren geplaatst maar nog niet eerder voor de Commodore 128. Het is het al om bekende Patience. Een uitleg wordt er in het programma gegeven, zodat ons allen nog rest de maker van het programma te vermelden. En dat is: B.A.D. van den Brom uit Noordwijk.

```
1       rem *** patience "harp[SPACE]9" /
        c 128 ***
2       rem *** door  b.a.d. van den brom
        ***
3       rem ***       noordwijk
        ***
10      fast:color6,12
20      scnclr:n=1:tl=0
30      gosub10000: rem * titelblad *
40      dima$(62),e$(62),ry$(11),mm$(15),s
        s$(15),g(15),tl(15)
50      for x=11 to 62: rem * kaarten-arra
        y vullen *
60      : reade$(x)
70      next
80      goto1000: rem * sprong over subrou
        tines naar begin spel *
100     rem ** controleren kaartvolgorde v
        an vp$tring **
110     for z=1 to lv-5 step3
120     : k=val(mid$(vp$,z,3))
130     : c=val(mid$(vp$,z+3,3))
140     : gosub200
150     next
160     return
200     rem ** controle op aansluiting kaa
        rt **
210     if k>11 and k<23 then begin
220     : if c=k+25 or c=k+38 thenreturn
230     bend
240     if k>24 and k<36 then begin
250     : if c=k+12 or c=k+25 thenreturn
260     bend
270     if k>37 and k<49 then begin
280     : if c=k-14 or c=k-27 thenreturn
290     bend
300     if k>50and k<62 then begin
310     : if c=k-27 or c=k-40thenreturn
320     bend
330     fout=1
340     return
400     rem ** foutmelding **
410     char,60,10,"[CTRL-3]dat[SPACE]gaat
        [SPACE]niet[COM-5]",1
420     sleep1
430     fout=0
440     return
500     rem ** schoonmaken schermdeel **
510     for z=7 to 10
520     : char,60,z,"[CTRL-0][19xSPACE]"
530     next
540     return
600     rem ** disk fout-routine **
610     if ds=63 then return
620     if ds>1 then printds$:stop
630     return
1000    fast:rem ** start spel **
1010    char,11,23,"[COM-5][SPACE]e[SPACE]
        v[SPACE]e[SPACE]n[3xSPACE]g[SPACE]
        e[SPACE]d[SPACE]u[SPACE]l[SPACE]d[
```

```
         SPACE]"
1020     for x=11 to 62: rem * hulp-array m
         et kaartnummers vullen *
1030     : a$(x)=str$(x)
1040     next
1050     for l=1 to 9: rem * kaartenstrings
          maken *
1060     : do
1070     :    x=int(rnd(0)*52)+11:ifa$(x)=""
         then1070
1080     :    ry$(l)=ry$(l)+a$(x):a$(x)=""
1090     :    if len(ry$(l))=30-(l*3) then e
         xit
1100     : loop
1110     next
1120     slow
1130     do: rem * laatste 7 kaarten in res
         t-ry$tring opslaan *
1140     : x=int(rnd(0)*52)+11:ifa$(x)=""th
         en1140
1150     : ry$(10)=ry$(10)+a$(x):a$(x)=""
1160     : if len(ry$(10))=21 then exit
1170     loop
1180     fast
1190     rem * scherm opbouwen *
1200     scnclr:color5,7
1210     for z=1 to 19:char,z*3,0,chr$(z+64
         ):next
1220     char,0,1,"[SHIFT-*]":char,1,1,"[SH
         IFT +]":for z=2 to 57 step3:char,z
         ,1,"[SHIFT-*][COMáR][SHIFT-*]":nex
         t
1230     char,1,0,"B":for z=2 to 19:char,1,
         z,"B":next
1240     for z=21 to23:char,1,z,"B":next
1250     char,0,20,"[SHIFT-*]":char,1,20,"[
         SHIFT-+]":for z=2 to 57 step3:char
         ,z,20,"[SHIFT-*][COM-E][SHIFT-*]":
         next
1260     char,59,0,"B":char,59,1,"[COM-W]":
         for z=2 to 19:char,59,z,"B":next
1270     char,59,20,"[COM-W]":for z=21 to 2
         3:char,59,z,"B":next
1280     for z=1 to 9 :char,0,z*2,chr$(z+48
         )
1290     char,31,z*2,chr$(z+48):next
1300     for z=1 to 19:char,z*3,8,chr$(z+64
         )
1310     char,z*3,14,chr$(z+64)
1320     char,z*3,21,chr$(z+64):next
1330     char,0,22,"x":for z=35 to 53 step6
         :char,z,22,"O[COM-Y]P":char,z,23,"
         L[COM-P][SHIFT-`]":next
1340     char,60,2,"geef[SPACE]de[SPACE]coo
         rdinaten"
1350     char,60,3,"om[2xSPACE]een[2xSPACE]
         kaart[2xSPACE]te"
1360     char,60,4,"verplaatsen[SPACE](bv[S
         PACE]3f)"
1370     char,60,5,"corrigeren[SPACE]met:[S
         PACE]< >"
1380     char,60,11,"spel[SPACE]beeindigen"
1390     char,60,12,"met[SPACE]:[SPACE]<hom
         e>"
1400     rem * kaarten-rijen neerleggen
1410     for z=1 to 9
1420     y=1
1430     : l=len(ry$(z))
1440     : do until y>l
1450     :    x=val(mid$(ry$(z),y,3))
1460     :    char,y+1,z*2,e$(x)
1470     :    y=y+3
1480     : loop
1490     next
1500     y=1
1510     l=len(ry$(10))
1520     do until y>l
1530     : x=val(mid$(ry$(10),y,3))
1540     : char,y+1,22,e$(x)
1550     : y=y+3
1560     loop
1570     printchr$(7):ti$="000000"
1580     gosub500:char,60,7,"[COM-5]van[SPA
         CE]waar[SPACE]?[2xSPACE]"
1590     getkeyx$:an$="123456789x[HOME]"
1600     ac=instr(an$,x$):if ac=0then 1590
1610     if ac=11 then char,60,7,"[COM-3]og
         enblikje",1:goto 2340:rem * spel b
         eeindigen *
1620     printmid$(an$,ac,1);"[SPACE]";
1630     getkeyx$:an$="abcdefghijklmnopqrs
         "
1640     al=instr(an$,x$):if al=0then 1630
1650     if al=20then 1580: rem * correcti
         e-mogelijkheid *
1660     printmid$(an$,al,1)
1670     rem * te verplaatsen kaart of kaar
         ten bepalen *
1680     if ac<10then begin
1690     : lr=len(ry$(ac))
1700     : lv=lr+3-(al*3)
1710     : if lv<1 then fout=1:goto1750
1720     : vp$=right$(ry$(ac),lv)
1730     : if lv>3 then gosub100: rem * con
         trole kaartvolgorde via kc$tring *
1740     bend
1750     if fout=1 thengosub400:goto1580
1760     if ac=10then vp$=mid$(ry$(10),al*
         3-2,3)
1770     x=val(left$(vp$,3))
1780     char,76,7,e$(x)
1790     char,60,9,"[CTRL-0][COM-5]naar[SPA
         CE]welke[SPACE]rij?[SPACE]"
1800     getkeyx$:an$="123456789x "
1810     bc=instr(an$,x$):if bc=0then 1800
1820     if bc=11 then 1580: rem * correcti
         e-mogelijkheid *
1830     printmid$(an$,bc,1)
1840     rem * controle aansluiting en verp
         laatsing kaart(en) *
1850     if bc<10then begin
1860     : vv$=right$(ry$(bc),3)
1870     : if vv$="" then1920: rem * bij le
         ge rij geen controle nodig *
1880     : k=val(vv$)
1890     : c=val(left$(vp$,3))
1900     : gosub200: rem * controle op aans
         luiting kaart *
1910     : if fout=1 then gosub400:goto1580
         :rem * foutmelding en terug naar v
         raag *
1920     : le=len(ry$(bc))
1930     : ry$(bc)=ry$(bc)+vp$
1940     : if ac<10then begin
1950     :    for z=1 to lv step3
1960     :       x=val(mid$(vp$,z,3))
1970     :       char,le+z+1,2*bc,e$(x)
1980     :       char,3*al+z-2,2*ac,"[CTRL-0]
         [3xSPACE][CRSR-DOWN][3xCRSR-LEFT][
         3xSPACE]"
1990     :    next
2000     :    ry$(ac)=left$(ry$(ac),lr-lv)
```

```
2010  :   if ac=4 or ac=7 then begin:pri
      nt"[CTRL-7]"
2020  :      lx=len(ry$(ac))/3+1
2030  :      for z=lx to 19
2040  :         char,z*3,ac*2,chr$(64+z)
2050  :      next
2060  :   bend
2070  : bend
2080  : if ac=10then begin
2090  :    x=val(vp$)
2100  :    char,le+2,2*bc,e$(x)
2110  :    char,3*al-1,22,"[CTRL-0][3xSPA
      CE][CRSR-DOWN][3xCRSR-LEFT][3xSPAC
      E]"
2120  :    ry$(10)=left$(ry$(10),3*(al-1)
      )+"[3xSPACE]"+right$(ry$(10),3*(7-
      al))
2130  : bend
2140  bend
2150  if bc=10then begin: rem * azen we
      gleggen met controle *
2160  : x=val(vp$)
2170  : if x=23 or x=36 or x=49 or x=62
      then2200:else begin
2180  :    gosub400:goto1580:rem * foutme
      lding en terug *
2190  : bend
2200  : le=len(ry$(11))
2210  : ry$(11)=ry$(11)+str$(x)+"[3xSPAC
      E]"
2220  : char,le+35,22,e$(x)
2230  : if ac<10then begin
2240  :    char,3*al-1,2*ac,"[CTRL-0][3xS
      PACE][CRSR-DOWN][3xCRSR-LEFT][3xSP
      ACE]"
2250  :    ry$(ac)=left$(ry$(ac),lr-3)
2260  : bend
2270  : if ac=10then begin
2280  :    char,3*al-1,22,"[CTRL-0][3xSPA
      CE][CRSR-DOWN][3xCRSR-LEFT][3xSPAC
      E]"
2290  :    ry$(10)=left$(ry$(10),3*(al-1)
      )+"[3xSPACE]"+right$(ry$(10),3*(7-
      al))
2300  : bend
2310  : tl=tl+1
2320  bend
2330  goto 1580: rem * terug voor volgen
      de kaartwissel *
2340  rem * aantal goed liggende kaarten
       tellen *
2350  fout=0
2360  for y=1 to 9
2370  : lv=len(ry$(y)):vp$=ry$(y)
2380  : if lv<6 then 2470
2390  : for z=lv-5 to 1 step -3
2400  :    k=val(mid$(vp$,z,3))
2410  :    c=val(mid$(vp$,z+3,3))
2420  :    gosub200: rem * controle aansl
      uiting kaarten *
2430  :    if fout=0 then tl=tl+1
2440  :    if fout=1 then 2460
2450  : next z
2460  : fout=0
2470  next y
2480  tl(n)=tl
2490  if tl=48 then begin:rem * spel uit
      gespeeld *
2500  : char,60,15,"[COM-3]alles[SPACE]l
      igt[SPACE]goed",1
2510  : g(n)=0
2520  bend
2530  if tl<48 then begin:rem * spel vas
      tgelopen *
2540  : char,60,15,"[CTRL-1]geen[SPACE]s
      ucces",1
2550  : g(n)=1
2560  bend
2570  mm$=mid$(ti$,3,2):ss$=mid$(ti$,5,2
      )
2580  mm$(n)=mm$:ss$(n)=ss$
2590  gosub500
2600  char,60,17,"[CTRL-8]gespeelde[SPAC
      E]tijd:"
2610  char,60,18,"[SPACE]":printmm$;"[SP
      ACE]minuten[SPACE]en"
2620  char,60,19,"[SPACE]":printss$;"[SP
      ACE]seconden"
2630  char,60,20,"behaalde[SPACE]punten:
      [COM-3]"
2640  char,60,21:printtl
2650  window0,0,59,24,1
2660  char,1,1,"[CTRL-8][2xSPACE]het[SPA
      CE]overzicht[SPACE]van[SPACE]de[SP
      ACE]spelresultaten[SPACE]:[SPACE]",1
2670  s=n:tt=0:gs=0:gt=0:pp=0
2680  for n=1 to s
2690  : char,1,n+2:print"[COM-5]spel"n"i
      n[SPACE]"mm$(n)"[SPACE]min[SPACE]e
      n[SPACE]"ss$(n)"[SPACE]sec[SPACE]m
      et"tl(n)"punten";
2700  : if g(n)=1 thenprint"[SPACE](afge
      br.)":else print
2710  : if g(n)=0then begin
2720  :    tt=tt+val(mm$(n))*60+val(ss$(n))
2730  :    gs=gs+1
2740  :    gt=int(tt/gs*100)/100:mm=int(g
      t/60):ss=int(gt-mm*60)
2750  : bend
2760  : pp=pp+tl(n)
2770  next
2780  gp=int(pp*10/s)/10
2790  print"[SPACE][CRSR-DOWN][CTRL-8][C
      TRL 9]gemiddelde[SPACE]speeltijd";
      gs;"[CRSR-LEFT][SPACE]volledige[SP
      ACE]spelen:[CTRL-0][CTRL-7]"
2800  printmm;"[SPACE]minuten[SPACE]en[S
      PACE]";ss;"[SPACE]seconden"
2810  print"[SPACE][CRSR-DOWN][CTRL-8][C
      TRL 9]gemiddelde[SPACE]aantal[SPAC
      E]punten[SPACE]van"s"[CRSR-LEFT][S
      PACE]spelen[SPACE]:[SPACE][CTRL-0]
      [COM-3]"gp
2820  window0,0,79,24
2830  char,60,23,"[COM-5]nog[SPACE]een[S
      PACE]spel[SPACE](j/n)?"
2840  getkeyx$:an$="jn"
2850  an=instr(an$,x$):if an=0then2840
2860  if an=1 then begin
2870  : for z=1 to 11
2880  :    ry$(z)=""
2890  : next
2900  : n=s+1:tl=0
2910  : scnclr
2920  : gosub10000:goto1000:rem * titelb
      lad en terug naar begin nieuw spel *
2930  bend
2940  char,0,23,"[COM-6]de[SPACE]spelres
      ultaten[SPACE]worden[SPACE]genotee
      rd"
2950  cr$=chr$(13):nm$="harp-9-punten"
2960  dopen#2,(nm$),w
```

```
2970  gosub 600
2980  if ds=63 thenbegin
2990  : dclose#2
3000  : append#2,(nm$),d0,u8
3010  bend
3020  gosub 600:rem * disk fout-routine
      *
3030  for n=1 to s
3040  : print#2,mm$(n);cr$;ss$(n);cr$;g(
      n);cr$;tl(n)
3050  : gosub 600
3060  next
3070  dclose#2
3080  char,0,23,"[COM-5]alle[SPACE]spelr
      esultaten[SPACE]tot[SPACE]nog[SPAC
      E]toe[SPACE]bekijken[SPACE](j/n)?"
3090  getkeyx$:an$="jn"
3100  an=instr(an$,x$):if an=0then3100
3110  scnclr
3120  if an=2 then end
3130  char,0,0,"BspelBspelduurB[SPACE]be
      haalde[SPACE]punten[4xSPACE][CTRL
      8]Q[COM-5][SPACE]=[SPACE]cumulatie
      f[SPACE]gemiddelde[3xSPACE]B"
3140  char,0,1,"B[SPACE]nr[SPACE]Bmin[2x
      SPACE]secB[CTRL-7][9xSPACE]10[8xSP
      ACE]20[8xSPACE]30[8xSPACE]40[7xSPA
      CE][COM-5]B"
3150  char,0,2,"[COM-Q]CCCC[SHIFT-+]CCCC
      CCCC[SHIFT-+]CCCCCCCC[SHIFT-+]CCC
      CCCCCC[SHIFT-+]CCCCCCCC[SHIFT-+]C
      CCCCCCC[SHIFT-+]CCCCCCCC[SHIFT-+]
      CCCCCCCCCCCCCC"
3160  window0,3,79,24
3170  dopen#2,"harp-9-punten"
3180  gosub 600
3190  n=1:z=0:tt=0:gs=0:af=0
3200  do
3210  : input#2,mm$,ss$,g,tl
3220  : gosub 600
3230  : if mm$="" then3400
3240  : mb=val(mm$):sb=val(ss$)
3250  : if g=1 then af=af+1
3260  : if g=0then begin
3270  :    ts=mb*60+sb
3280  :    gs=gs+1
3290  :    tt=tt+ts
3300  : bend
3310  : print"B";tab(1)n;tab(5)"B";tab(6
      )mb;tab(10)sb;tab(14)"B";
3320  : p=p+tl
3330  : tm=int(p*10/n)/10
3340  : for x=15 to tl+14
3350  :    printtab(x)"[SHIFT-*]";
3360  : next
3370  : printtab(63)"B";:if g=1 then pri
      nt"afgebr.spel":else print
3380  : printtab(tm+14)"[CRSR-UP][CTRL-8
      ]Q[COM-5]"
3390  : z=z+1
3400  : if mm$="" then n=n-1:exit
3410  : if st<>64then printst:exit
3420  : n=n+1:mm$=""
3430  : if z=16 then begin
3440  :    print"[COM-3][SPACE]doorgaan[S
      PACE]?[SPACE]druk[SPACE]een[SPACE]
      toets[SPACE]in[COM-5]"
3450  :    getkeyx$:z=0
3460  :    print"[CRSR-UP][29xSPACE][CRSR
      -UP]"
3470  : bend
3480  loop
3490  dclose#2
3500  print:print"[CTRL-8][SPACE]gemidde
      ld:";
3510  for x=15 to tm+14
3520  : printtab(x)"[SHIFT-*]";
3530  next
3540  print"[SPACE]=[SPACE]"tm"[SPACE]pu
      nten"
3550  gt=int(tt/gs*100)/100:mm=int(gt/60
      ):ss=int(gt-mm*60)
3560  print:print"[CTRL-8][SPACE]gemidde
      lde[SPACE]speeltijd";gs;"[CRSR-LEF
      T][SPACE]volledige[SPACE]spelen:";
3570  printmm;"minuten[SPACE]en[SPACE]";
      ss;"seconden[SPACE]=";int(tt/gs);"
      sec.[CTRL-7]"
3580  print"[CTRL-7][CRSR-DOWN][SPACE]to
      taal[SPACE]gespeeld"n"[CRSR-LEFT][
      SPACE]spelen,[SPACE]waarvan"af"[CR
      SR-LEFT][SPACE]spelen[SPACE]zijn[S
      PACE]afgebroken"
3590  window0,0,79,24:sleep5
3600  char,1,24:end
10000 rem ** titelblad **
10010 color6,16:color5,5
10020 char,12,4,"[COM-+][15xCOM-|][COM-+
      ]"
10030 char,12,5,"[COM-+][SPACE][COM—][S
      PACE][COM—][SPACE][COM—][SPACE][
      COM—][SPACE][COM—][SPACE][COM—]
      [SPACE][COM—][SPACE][COM-+]"
10040 char,12,6,"[COM-+][SPACE][COM—][S
      PACE][COM—][SPACE][COM—][SPACE][
      COM—][SPACE][COM—][SPACE][COM—]
      [SPACE][COM—][SPACE][COM-+]"
10050 char,12,7,"[COM-+][SPACE][COM—][S
      PACE][COM—][SPACE][COM—][SPACE][
      COM—][SPACE][COM—][SPACE][COM—]
      [SPACE][COM—][COM-|][COM-+]"
10060 char,12,8,"[COM-+][SPACE][COM—][S
      PACE][COM—][SPACE][COM—][SPACE][
      COM—][SPACE][COM—][SPACE][COM—]
      [COM-|][COM-+]"
10070 char,12,9,"[COM-+][SPACE][COM—][S
      PACE][COM—][SPACE][COM—][SPACE][
      COM—][SPACE][COM—][COM-|][COM-+]"
10080 char,12,10,"[COM-+][SPACE][COM—][
      SPACE][COM—][SPACE][COM—][SPACE]
      [COM—][COM-|][COM-+]"
10090 char,12,11,"[COM-+][SPACE][COM—][
      SPACE][COM—][SPACE][COM—][COM-|]
      [COM-+][10xSPACE][CTRL-8][SPACE][S
      HIFT -][4xSPACE][SHIFT—]"
10100 char,12,12,"[CTRL-5][COM-+][SPACE]
      [COM—][SPACE][COM—][COM-|][COM-+
      ][12xSPACE][CTRL-8][SPACE][SHIFT—
      ][4xSPACE][SHIFT—][2xSPACE]U[2xSH
      IFT *]I[2xSPACE][COM-A][2xSHIFT-*]
      I[2xSPACE][COM-A][2xSHIFT-*]I[5xSP
      ACE]U[3xSHIFT-*]I"
10110 char,12,13,"[CTRL-5][COM-+][SPACE]
      [COM—][COM-|][COM-+][14xSPACE][CT
      RL 8][SPACE][COM-Q][4xSHIFT-*][COM
       W][2xSPACE][SHIFT—][2xSPACE][SHI
      FT -][2xSPACE][SHIFT—][2xSPACE][S
      HIFT -][2xSPACE][SHIFT—][2xSPACE]
      [SHIFT—][5xSPACE][SHIFT—][3xSPAC
      E][SHIFT—]"
10120 char,12,14,"[CTRL-5][COM-+][COM-|]
      [COM-+][16xSPACE][CTRL-8][SPACE][S
```

```
       HIFT -][4xSPACE][SHIFT—][2xSPACE]
       [COM-Q][2xSHIFT-*][COM-W][2xSPACE]
       [COM-Q][SHIFT-*][COMáR]K[2xSPACE][
       COM-Q][2xSHIFT-*]K[5xSPACE]J[3xSHI
       FT *][COM-W]"
10130  char,12,15,"[CTRL-5][COM-+][18xSPA
       CE][CTRL-8][SPACE][SHIFT—][4xSPAC
       E][SHIFT—][2xSPACE][SHIFT—][2xSP
       ACE][SHIFT—][2xSPACE][SHIFT—][SP
       ACE]JI[2xSPACE][SHIFT—][10xSPACE]
       [2xSHIFT-*]K"
10140  char,31,19,"[COM-3]een[SPACE]patie
       nce-spel[SPACE]met[SPACE]9[SPACE]r
       ijen[SPACE]kaarten"
10150  return
25000  rem ** data van de kaarten **
25010  data"[CTRL-9][CTRL-3][SPACE]2[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]S
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]3[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]S[SPACE][CRSR-UP]"
25020  data"[CTRL-9][CTRL-3][SPACE]4[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]S
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]5[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]S[SPACE][CRSR-UP]"
25030  data"[CTRL-9][CTRL-3][SPACE]6[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]S
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]7[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]S[SPACE][CRSR-UP]"
25040  data"[CTRL-9][CTRL-3][SPACE]8[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]S
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]9[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]S[SPACE][CRSR-UP]"
25050  data"[CTRL-9][CTRL-3]1[SPACE]0[CRS
       R-DOWN][3xCRSR-LEFT][SPACE]S[SPACE
       ][CRSR-UP]","[CTRL-9][CTRL-3]TbY[C
       RSR-DOWN][3xCRSR-LEFT]TSY[CRSR-UP]"
25060  data"[CTRL-9][CTRL-3]TvY[CRSR-DOWN
       ][3xCRSR-LEFT]TSY[CRSR-UP]","[CTRL
        9][CTRL-3]ThY[CRSR-DOWN][3xCRSR-L
       EFT]TSY[CRSR-UP]"
25070  data"[CTRL-9][CTRL-2]TaY[CRSR-DOWN
       ][3xCRSR-LEFT]TSY[CRSR-UP]","[CTRL-
     9][CTRL-3][SPACE]2[SPACE][CRSR-DO
       WN][3xCRSR-LEFT][SPACE]Z[SPACE][CR
       SR-UP]"
25080  data"[CTRL-9][CTRL-3][SPACE]3[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]Z
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]4[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]Z[SPACE][CRSR-UP]"
25090  data"[CTRL-9][CTRL-3][SPACE]5[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]Z
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]6[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]Z[SPACE][CRSR-UP]"
25100  data"[CTRL-9][CTRL-3][SPACE]7[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]Z
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ][SPACE]8[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]Z[SPACE][CRSR-UP]"
25110  data"[CTRL-9][CTRL-3][SPACE]9[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]Z
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-3
       ]1[SPACE]0[CRSR-DOWN][3xCRSR-LEFT]
       [SPACE]Z[SPACE][CRSR-UP]"
25120  data"[CTRL-9][CTRL-3]TbY[CRSR-DOWN
       ][3xCRSR-LEFT]TZY[CRSR-UP]","[CTRL
        9][CTRL-3]TvY[CRSR-DOWN][3xCRSR-L
       EFT]TZY[CRSR-UP]"
25130  data"[CTRL-9][CTRL-3]ThY[CRSR-DOWN
       ][3xCRSR-LEFT]TZY[CRSR-UP]","[CTRL
        9][CTRL-2]TaY[CRSR-DOWN][3xCRSR-L
       EFT]TZY[CRSR-UP]"
25140  data"[CTRL-9][CTRL-1][SPACE]2[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]A
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]3[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]A[SPACE][CRSR-UP]"
25150  data"[CTRL-9][CTRL-1][SPACE]4[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]A
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]5[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]A[SPACE][CRSR-UP]"
25160  data"[CTRL-9][CTRL-1][SPACE]6[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]A
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]7[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]A[SPACE][CRSR-UP]"
25170  data"[CTRL-9][CTRL-1][SPACE]8[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]A
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]9[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]A[SPACE][CRSR-UP]"
25180  data"[CTRL-9][CTRL-1]1[SPACE]0[CRS
       R-DOWN][3xCRSR-LEFT][SPACE]A[SPACE
       ][CRSR-UP]","[CTRL-9][CTRL-1]TbY[C
       RSR-DOWN][3xCRSR-LEFT]TAY[CRSR-UP]"
25190  data"[CTRL-9][CTRL-1]TvY[CRSR-DOWN
       ][3xCRSR-LEFT]TAY[CRSR-UP]","[CTRL
        9][CTRL-1]ThY[CRSR-DOWN][3xCRSR-L
       EFT]TAY[CRSR-UP]"
25200  data"[CTRL-9][CTRL-2]TaY[CRSR-DOWN
       ][3xCRSR-LEFT]TAY[CRSR-UP]","[CTRL
        9][CTRL-1][SPACE]2[SPACE][CRSR-DO
       WN][3xCRSR-LEFT][SPACE]X[SPACE][CR
       SR-UP]"
25210  data"[CTRL-9][CTRL-1][SPACE]3[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]X
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]4[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]X[SPACE][CRSR-UP]"
25220  data"[CTRL-9][CTRL-1][SPACE]5[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]X
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]6[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]X[SPACE][CRSR-UP]"
25230  data"[CTRL-9][CTRL-1][SPACE]7[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]X
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ][SPACE]8[SPACE][CRSR-DOWN][3xCRSR
       -LEFT][SPACE]X[SPACE][CRSR-UP]"
25240  data"[CTRL-9][CTRL-1][SPACE]9[SPAC
       E][CRSR-DOWN][3xCRSR-LEFT][SPACE]X
       [SPACE][CRSR-UP]","[CTRL-9][CTRL-1
       ]1[SPACE]0[CRSR-DOWN][3xCRSR-LEFT]
       [SPACE]X[SPACE][CRSR-UP]"
25250  data"[CTRL-9][CTRL-1]TbY[CRSR-DOWN
       ][3xCRSR-LEFT]TXY[CRSR-UP]","[CTRL
        9][CTRL-1]TvY[CRSR-DOWN][3xCRSR-L
       EFT]TXY[CRSR-UP]"
25260  data"[CTRL-9][CTRL-1]ThY[CRSR-DOWN
       ][3xCRSR-LEFT]TXY[CRSR-UP]","[CTRL
        9][CTRL-2]TaY[CRSR-DOWN][3xCRSR-L
       EFT]TXY[CRSR-UP]"
```

EINDE LISTING patience

Checksum patience

```
REGEL 1     137
REGEL 2     252
REGEL 3     171
REGEL 10    9
REGEL 20    15
REGEL 30    48
REGEL 40    155
REGEL 50    158
REGEL 60    211
REGEL 70    130
REGEL 80    154
REGEL 100   138
REGEL 110   192
REGEL 120   23
REGEL 130   236
REGEL 140   89
REGEL 150   130
REGEL 160   142
REGEL 200   238
REGEL 210   184
REGEL 220   80
REGEL 230   23
REGEL 240   192
REGEL 250   72
REGEL 260   23
REGEL 270   200
REGEL 280   78
REGEL 290   23
REGEL 300   190
REGEL 310   77
REGEL 320   23
REGEL 330   33
REGEL 340   142
REGEL 400   117
REGEL 410   166
REGEL 420   58
REGEL 430   32
REGEL 440   142
REGEL 500   73
REGEL 510   201
REGEL 520   52
REGEL 530   130
REGEL 540   142
REGEL 600   243
REGEL 610   114
REGEL 620   201
REGEL 630   142
REGEL 1000  86
REGEL 1010  234
REGEL 1020  81
REGEL 1030  103
REGEL 1040  130
REGEL 1050  70
REGEL 1060  37
REGEL 1070  221
REGEL 1080  186
REGEL 1090  21
REGEL 1100  38
REGEL 1110  130
REGEL 1120  36
REGEL 1130  182
REGEL 1140  219
REGEL 1150  228
REGEL 1160  3
REGEL 1170  236
REGEL 1180  35
REGEL 1190  20
REGEL 1200  161
REGEL 1210  21
REGEL 1220  40
REGEL 1230  189
REGEL 1240  228
REGEL 1250  186
REGEL 1260  107
REGEL 1270  134
REGEL 1280  41
REGEL 1290  68
REGEL 1300  97
REGEL 1310  136
REGEL 1320  66
REGEL 1330  54
REGEL 1340  22
REGEL 1350  193
REGEL 1360  233
REGEL 1370  44
REGEL 1380  110
REGEL 1390  52
REGEL 1400  64
REGEL 1410  155
REGEL 1420  60
REGEL 1430  198
REGEL 1440  119
REGEL 1450  211
REGEL 1460  28
REGEL 1470  123
REGEL 1480  38
REGEL 1490  130
REGEL 1500  60
REGEL 1510  147
REGEL 1520  61
REGEL 1530  218
REGEL 1540  72
REGEL 1550  123
REGEL 1560  236
REGEL 1570  249
REGEL 1580  136
REGEL 1590  65
REGEL 1600  87
REGEL 1610  131
REGEL 1620  46
REGEL 1630  214
REGEL 1640  100
REGEL 1650  247
REGEL 1660  125
REGEL 1670  198
REGEL 1680  224
REGEL 1690  66
REGEL 1700  113
REGEL 1710  163
REGEL 1720  66
REGEL 1730  126
REGEL 1740  23
REGEL 1750  5
REGEL 1760  181
REGEL 1770  98
REGEL 1780  26
REGEL 1790  47
REGEL 1800  141
REGEL 1810  83
REGEL 1820  239
REGEL 1830  117
REGEL 1840  2
REGEL 1850  225
REGEL 1860  218
REGEL 1870  232
REGEL 1880  29
REGEL 1890  135
REGEL 1900  45
REGEL 1910  71
REGEL 1920  54
REGEL 1930  170
REGEL 1940  26
REGEL 1950  26
REGEL 1960  36
REGEL 1970  131
REGEL 1980  11
REGEL 1990  188
REGEL 2000  100
REGEL 2010  63
REGEL 2020  3
REGEL 2030  121
REGEL 2040  191
REGEL 2050  188
REGEL 2060  81
REGEL 2070  81
REGEL 2080  25
REGEL 2090  36
REGEL 2100  128
REGEL 2110  8
REGEL 2120  202
REGEL 2130  81
REGEL 2140  23
REGEL 2150  12
REGEL 2160  36
REGEL 2170  48
REGEL 2180  97
REGEL 2190  81
REGEL 2200  19
REGEL 2210  245
REGEL 2220  183
REGEL 2230  26
REGEL 2240  6
REGEL 2250  245
REGEL 2260  81
REGEL 2270  25
REGEL 2280  8
REGEL 2290  202
REGEL 2300  81
REGEL 2310  7
REGEL 2320  23
REGEL 2330  223
REGEL 2340  188
REGEL 2350  32
REGEL 2360  154
REGEL 2370  74
REGEL 2380  196
REGEL 2390  165
REGEL 2400  23
REGEL 2410  236
REGEL 2420  33
REGEL 2430  89
REGEL 2440  89
REGEL 2450  22
REGEL 2460  90
REGEL 2470  219
REGEL 2480  145
REGEL 2490  146
REGEL 2500  97
REGEL 2510  2
REGEL 2520  23
REGEL 2530  160
REGEL 2540  128
REGEL 2550  3
REGEL 2560  23
REGEL 2570  90
REGEL 2580  236
REGEL 2590  34
REGEL 2600  7
REGEL 2610  58
REGEL 2620  227
REGEL 2630  64
REGEL 2640  116
REGEL 2650  45
REGEL 2660  135
REGEL 2670  64
REGEL 2680  169
REGEL 2690  247
REGEL 2700  224
REGEL 2710  74
REGEL 2720  148
REGEL 2730  251
REGEL 2740  187
REGEL 2750  81
REGEL 2760  21
REGEL 2770  130
REGEL 2780  10
REGEL 2790  68
REGEL 2800  20
REGEL 2810  198
REGEL 2820  210
REGEL 2830  82
REGEL 2840  145
REGEL 2850  108
REGEL 2860  186
REGEL 2870  254
REGEL 2880  170
REGEL 2890  188
REGEL 2900  36
REGEL 2910  34
REGEL 2920  75
REGEL 2930  23
REGEL 2940  6
REGEL 2950  110
REGEL 2960  31
REGEL 2970  35
REGEL 2980  250
REGEL 2990  156
REGEL 3000  48
REGEL 3010  23
REGEL 3020  252
REGEL 3030  169
REGEL 3040  168
REGEL 3050  93
REGEL 3060  130
REGEL 3070  98
REGEL 3080  139
REGEL 3090  145
REGEL 3100  98
REGEL 3110  232
REGEL 3120  37
REGEL 3130  175
REGEL 3140  177
REGEL 3150  4
REGEL 3160  213
REGEL 3170  104
REGEL 3180  35
REGEL 3190  196
REGEL 3200  235
REGEL 3210  15
REGEL 3220  93
REGEL 3230  231
REGEL 3240  176
REGEL 3250  49
REGEL 3260  171
REGEL 3270  115
REGEL 3280  251
REGEL 3290  141
REGEL 3300  81
REGEL 3310  23
REGEL 3320  214
REGEL 3330  249
REGEL 3340  126
REGEL 3350  54
REGEL 3360  188
REGEL 3370  109
REGEL 3380  43
REGEL 3390  123
REGEL 3400  113
REGEL 3410  72
REGEL 3420  81
REGEL 3430  245
REGEL 3440  3
REGEL 3450  198
REGEL 3460  57
REGEL 3470  81
REGEL 3480  236
REGEL 3490  98
REGEL 3500  66
REGEL 3510  69
REGEL 3520  54
REGEL 3530  130
REGEL 3540  217
REGEL 3550  129
REGEL 3560  126
REGEL 3570  40
REGEL 3580  139
REGEL 3590  74
REGEL 3600  137
REGEL 10000  204
REGEL 10010  103
REGEL 10020  99
REGEL 10030  144
REGEL 10040  145
REGEL 10050  58
REGEL 10060  95
REGEL 10070  132
REGEL 10080  208
REGEL 10090  77
REGEL 10100  252
REGEL 10110  191
REGEL 10120  209
REGEL 10130  93
REGEL 10140  43
REGEL 10150  142
REGEL 25000  197
REGEL 25010  144
REGEL 25020  148
REGEL 25030  152
REGEL 25040  156
REGEL 25050  40
REGEL 25060  125
REGEL 25070  232
REGEL 25080  160
REGEL 25090  164
REGEL 25100  168
REGEL 25110  211
REGEL 25120  133
REGEL 25130  95
REGEL 25140  84
REGEL 25150  88
REGEL 25160  92
REGEL 25170  96
REGEL 25180  236
REGEL 25190  65
REGEL 25200  72
REGEL 25210  132
REGEL 25220  136
REGEL 25230  140
REGEL 25240  183
REGEL 25250  105
REGEL 25260  20
```

Het seizoen vordert weer. De lente is weer in het land. Het humeur van de mensen wordt beter naarmate de temperatuur stijgt en de zon langer schijnt. Ook in ons selecte computerwereldje blijft de invloed van de lente niet onopgemerkt. Ineens krijgt men weer zin om allerlei zorgvuldig gekoesterde programmeerdromen daadwerkelijk om te zetten naar een programma. Wat is er dan een betere informatiebron dan de TIPS en TRUCS hoek!! Fasten your seatbelts, get your keyboard, hier zijn we weer met.....

Aanraders & meepikkers

Ook deze keer hopen we voor ieder weer wat wils te hebben. Het blijft desondanks nog steeds roeien met de riemen die we hebben want voor deze hoek kunnen we nog veel meer reacties gebruiken. Zijn het tot nu toe tips en trucs die door ons zelf zijn verzameld, de tijd nadert snel dat we UW HULP NODIG HEBBEN!!!! Stuur ze daarom op al die kleine ideetjes, programma's, speltips, etcetera. We kunnen alle programma's, subroutines gebruiken. Het maakt niet uit in welke programmeertaal het is geschreven. Originaliteit staat voorop.



Tips en Trucs Wanted!!

Stuur ze op naar Commodore Info, Postbus 43048, 1009 ZA, Amsterdam, en vermeldt er bij 'Amiga Tips & Trucs'.

De mediterende Guru

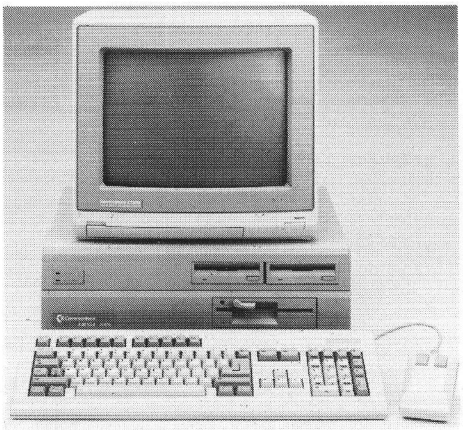
Meteen als goede start van deze aflevering een inslaande bom. Voor al de programmeurs in spé met een smalle beurs (de boeken zijn zo duur he!) hebben we hier een overzicht van de Guru-meditation foutcodes. U kent ze ongetwijfeld wel, DE grote angst van de Amiga bezitters, de zwarte schermen met bovenin een knipperende rode rand en een melding. Echter, het is niet een prettig geschreven foutmelding, nee, bureaucratie ten top, alleen maar cijfers. Voor beginners een totale wanhoop, maar ook gevorderden kunnen hiermee nog behoorlijk met hun handen in hun haar komen te zitten.

Waar gaat het nu om. Elke GURU-Meditation bestaat uit een stukje tekst en twee getallen gescheiden door een punt. Deze twee getallen zijn de sleutels tot het geheim. Door nu op te zoeken welke fout ze voorstellen kan men nagaan wat er ongeveer misgegaan is.

Wat is nu het formaat:

```
Guru Meditation : (TT)AABBCCCC.DDDDDDDD
```

TT stelt het zogenaamde 'ALERT_TYPE' voor. Elke Guru-meditation kan een zogenaamde RECO-



VERY_ALERT zijn, het is hierbij niet nodig om opnieuw te booten. Dit booten of eigenlijk resetten van de Amiga gebeurt wel. Dit is te wijten aan een 'bug' in de KickStart 1.2 ROM van de Amiga. Volgens de berichten van Commodore moet deze fout in KickStart 1.3 verholpen zijn. De Guru-meditation kan ook van het DEADEND_ALERT type zijn. Hierbij gaat de Amiga zondermeer door zijn reset- of bootfase heen. Hoe is dit fouttype nu te herkennen:

```
TT =
-----------------------------          00
  :  RECOVERY_ALERT
80  :  DEADEND_ALERT
-----------------------------
```

Dit ALERT_TYPE is niet zondermeer terug te vinden in de foutcode. Door nu het ALERT_TYPE (=TT) te OR'en met de AA, de twee eerste digits van de foutcode, krijgt men de werkelijke foutcode die op het scherm staat.

AA is de zogenaamde 'SUBSYSTEM_CODE'. Het is een wat nauwkeurigere aanduiding van in wat voor onderdeel van de soft- of hardware van de Amiga de fout is opgetreden. Dit zijn er een aantal.

```
AA=
------------------------------------------
      00   :   CPU 68000

libraries----------------------
      01   :   exec.library
      02   :   graphics.library
      03   :   layers.library
      04   :   intuition.library
      05   :   math.library
      06   :   clist.library
      07   :   dos.library
```



08 : ram.library
09 : icon.library
0A : expansion.library
devices----------------------
10 : audio.device
11 : console.device
12 : gameport.device
13 : keyboard.device
14 : trackdisk.device
15 : timer.device
resources---------------------
20 : CIA's (parallele- en seriele poorten)
21 : diskdrive
22 : diversen
overige bronnen----------------
30 : Bootstrap
31 : WorkBench
32 : Disk-Copy

De volgende group (BB) betreft een groep van algemene fouten.

BB=

---

00 :Fout kan niet in een klasse onderverdeeld worden
01 :Te weinig geheugen
02 :Een Library kon niet worden gegenereerd (MakeLibrary error)
03 :Een Library kon niet worden geopend (OpenLibrary error)
04 :Device kon niet geopend worden (OpenDevice error)
05 :Hardware onderdeel reageert niet (OpenResource error)
06 :I/O error
07 :I/O niet aanwezig

Tenslotte is er dan onderdeel CCCC. Dit geeft een nauwkeurige aanduiding van de fout. We geven een compleet overzicht per onderdeel. Eerst de trapcodes van de processor, vervolgens de library fouten, daarna de device fouten, de resource fouten volgen daarop en tenslotte de overige fouten. De gehele cijfercombinatie AABBCCDDDD wordt nu afgebeeld. Dus als u wilt zien of het een DEADEND of RECOVERY_ALERT betreft, kijk naar het eerste cijfer van AA. Als deze een 8 is dan gaat het om een

DEADEND_ALERT

AABBCCDDDD =

---

processor_traps----------------
00000002:Bus Error, er is een fout in de databus of de adresbus.
00000003:Address Error, een oneven adres is er op de adresbus gezet.
00000004:Illegal Instruction, een niet voor de MC68000 bestaande instructie.
00000005:Division by zero, een deling door nul.
00000006:CHK instructie, ze test of een register tegen zijn uiterste getal bereik komt. Als de grenzen overschreden worden dan treedt de fout op.
00000007:TRAPV instructie. Deze fout wordt gegenereerd als bij gezette V bit de TRAPV instructie wordt gebruikt.
00000008:Privelege Violation. Een zogenaamd 'priveledged instruction' is gebruikt.
00000009:Trace mode. Stap voor stap modus.
0000000A:OP Code 1010. Ongebruikte opcode.
0000000B:OP Code 1111. Ongebruikte opcode.

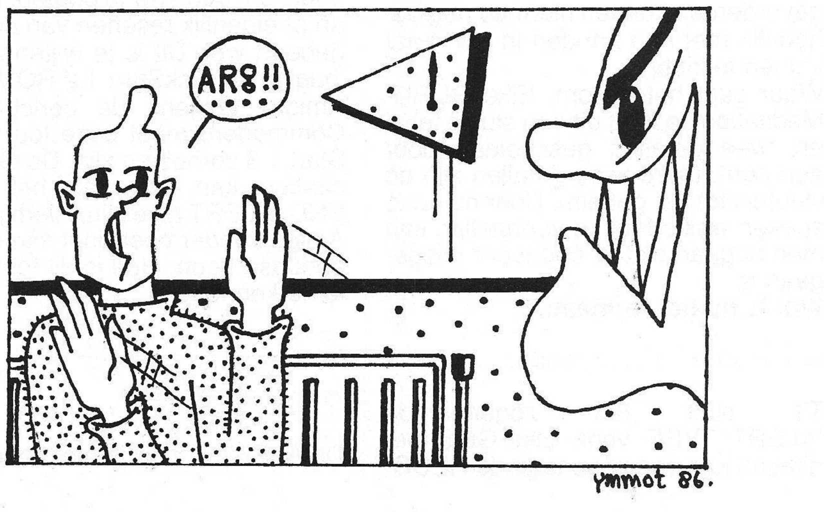
exec.library----------------
81000001:checksum-fout bij een exception van de MC68000
81000002:checksum-fout van ExecBase adres.
81000003:checksum-fout bij een library. In de praktijk betekent dit vaak een onvolledige of kortweg gezegd een kapotte library.
81000004:Geen geheugen genoeg om een library te maken.
81000005:een onjuiste boeking in de geheugen lijst (MemList).
81000006:geen geheugen genoeg voor Interrupt Servers.
81000007:verkeerde pointer.
81000008:Semaphore niet juist.
81000009:twee keer de zelfde geheugenblokken vrij gegeven. FreeMem()
8100000A:een nog niet bestaande Exception vector werd aangeroepen.

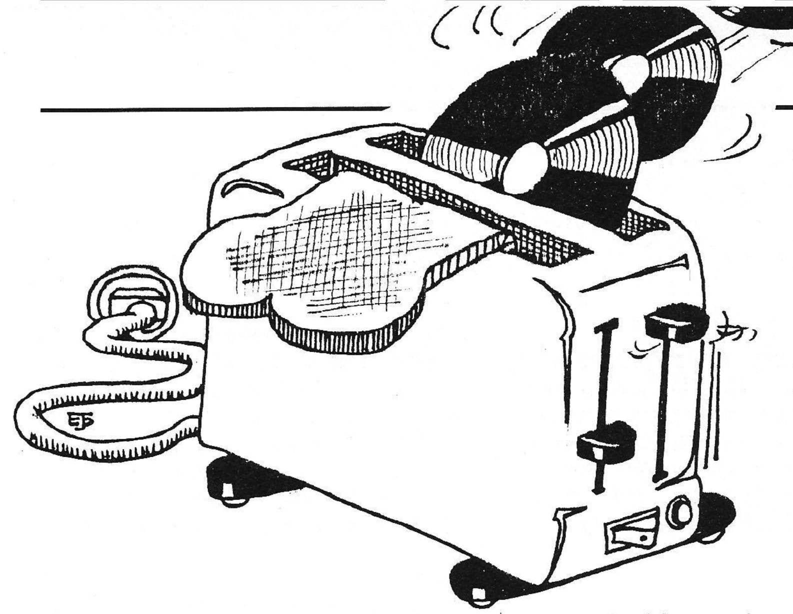
graphics.library-------------
82010001:geen geheugen voor een Copper-(Copper Display) lijst.
82010002:geen geheugen genoeg voor de Copper Instructie Lijst.
82000003:de Copper Lijst is vol.
82000004:indelingsfout in de Copper Lijst.
82010005:geen geheugen genoeg voor de Copper Lijst Header.
82010006:geen geheugen genoeg voor de Long Frame.
82010007:geen geheugen genoeg voor de Short Frame.
82010008:geen geheugen genoeg voor de Flood() instructie.
82010009:geen geheugen genoeg voor de Text() instructie.
8201000A:geen geheugen genoeg voor de BltBitMap() instructie.
8201000B:verkeerd geheugengebied aangeroepen.
82010030:er is een fout opgetreden bij het aanmaken van een nieuwe ViewPort (MakeVPort).
82011234:GfxNoLCM (geen grafische buffer beschikbaar).

layers.library---------------
83010001:geen geheugen genoeg voor voor Layers (LayersNoMem).

intuition.library------------
84000001:een onbekend Gadget type.
84010002:geen geheugen genoeg om een Port aan te maken.
84010003:geen geheugen genoeg om een Menu te maken.
84010004:geen geheugen genoeg om een SubMenu te maken.
84010005:geen geheugen genoeg voor de MenuBalk.
84000006:de positie van de MenuBalk is verkeerd oftewel, onder het nulpunt van het scherm.
84010007:geen geheugen genoeg voor OpenScreen()
84010008:geen geheugen genoeg voor het maken van een RastPort
84000009:onbekend of verkeerd schermtype in SCREEN_TYPE
8401000A:geen geheugen genoeg voor Gadget
8401000B:geen geheugen genoeg voor OpenWindow()
8400000C:verkeerde status in het statusregister bij opening van de 'intuition.library'
8400000D:verkeerde Message via de IDCMP
8400000E:niet genoeg geheugen voor de Message stack
8400000F:het Console Device kon niet geopend worden

dos.library-----------------
07010001:geen geheugen genoeg bij de Startup
07000002:het commando EndTask() vervulde niet goed zijn taak, de Task is niet beeindigd
07000003:fout in QueuePacket. Data niet goed overgedragen
07000004:er is een datapakket aangekomen die niet verwacht was
07000005:het commando FreeVec() heeft niet goed gewerkt
07000006:er is een fout in de gegevens van het gelezen Disk-blok
07000007:foutieve of beschadigde BitMap
07000008:het File-nummer (Key) is al vrijgegeven
07000009:verkeerde Checksum
0700000A:Disk Error
0700000B:File-nummer (Key-nummer) buiten het bereik
0700000C:'Overlay-Hunk' in de Bootsector is foutief





ram.library-----------------
08000001:Bad Segment List, een foutieve boeking in de geheugenlijst

expansion.library-----------
0A000001:er is een fout geconstateerd in de aan de Amiga gekoppelde Hardware

trackdisk.device-----------
14000001:Calibrate: Seek Error, er is een fout geconstateerd bij de zoekoperoperatie op de schijf
14000002:er is een fout opgetreden tijdens het wachten op een Timer impuls

timer.device---------------
15000001:er is een fout opgetreden tijdens het gebruik van het Timer Device
15000002:een 'smerige' netspanning geeft problemen bij het Timer Device

disk-resource--------------
21000001:de geplaatste diskette is niet herkenbaar
21000002:interrupt: er is (zijn) geen actieve loopwerk(en) gekoppeld aan de Amiga

BootStrap------------------
30000001:er is een foutmelding gekomen terwijl de dos.library tijdens de bootfase nog niet geopend was

Dit waren alle specifieke foutcodes. Dan rest ons nog één blok, het blok DDDDDDDD. Hierin is het adres vermeldt waar de fout opgetreden is. Nu is het slechts nog een kwestie van debuggen van de software. Heel vaak zult u tijdens het ontwikkelen van de software de codes '00000003' en '00000004' tegenkomen. Hierbij is het, in bijvoorbeeld een C programma, niet direct mogelijk een bepaald programmadeel als foutief aan te merken. Toch kunt u dan met enig logisch redeneerwerk de fout wel vinden. Hiervoor zijn geen echte logische paden waarmee u de fout kunt vinden. Ervaring brengt inzicht met zich mee, zullen we maar zeggen.

Onzichtbare pointer

Een BASIC-listing nu weer. Er zijn van die dagen dat u denkt, wat moet ik in vredesnaam met die walgelijke muispointer. Kunt u zich hierin vinden? Goed, de oplossing voor dit probleem is hier onder vermeldt. Het principe komt neer op de aard van de pointer. De muispointer is de eerste van de hardware-sprites. Deze sprites kunt u afzonderlijk aan of uit schakelen. Hiervoor levert de 'graphics.library' een functie genaamd 'FreeSprite(n)'. Roep deze aan met in 'n' het nummer van de sprite vermeldt en tataaa, weg is de sprite!

```
LIBRARY "graphics.library"

CALL FreeSprite(0)
```

Dit kleine routinetje zorgt ervoor dat de muispointer 'history' is.
Hoe krijg ik 'm weer terug....resetten?, vraagt u zich af. Beweeg de muis eens. Daar is ie weer!!!



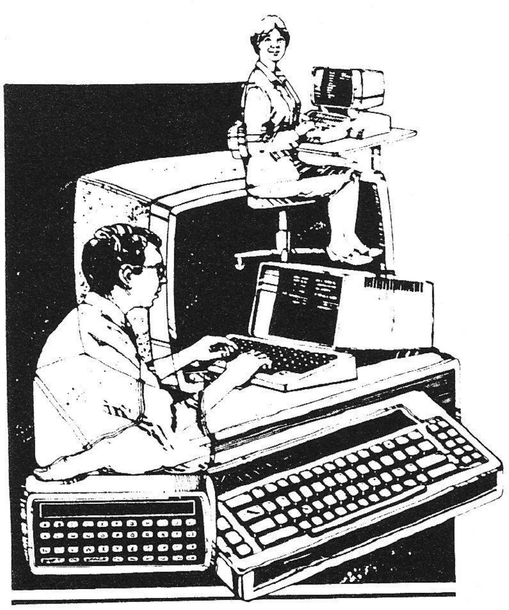
CIA, FBI, KGB, VIA

Bovenstaand kopje doet vermoeden dat het verhaal een begin is van een spannende spionage thriller. Helaas we gaan het niet over de 'Central Intelligence Agency' hebben maar over de 'Complex Interface Adapters' in de Amiga. Hardware freaks kennen dezen beter als de 8520(A)'s. Deze twee IC's regelen alle seriële- en parallele in- en uitvoer. Hoe zijn dezen nu te programmeren. Het antwoord luidt in vele gevallen; gewoon rechtstreeks beinvloeden. Vanuit BASIC kunt u getallen gaan 'poken', in C kent u aan een basisvariabele het adres toe en u kunt beginnen. Hoe ziet de geheugenkaart van de CIA's er uit?

CIA-A:
$BFE001:Poort A, bedoelt voor meerdere toepassingen.
bit7 :Vuurknop 1
bit6 :Vuurknop 0
bit5 :RDY, ready-signaal van de diskdrive(s)
bit4 :TK0, track 00-signaal van de diskdrive(s)
bit3 :WPRO, write-protect signaal van de diskdrive(s)
bit2 :CHNG, diskchange signaal van de diskdrive(s)
bit1 :LED, besturing van de powerled (0=uit). In vele Amigamodellen betekent dit ook besturing van het Highpass filter.
bit0 :OVL, Memory OverLay bit. Ten alle tijden niet wijzigen. Een 1 betekent dat er kopie van het KickStart ROM op $00000000 komt te staan.
$BFE101:Poort B, de centronix poort. Vrij programmeerbaar als u geen paralelle toepassingen heeft.
$BFE201:Data direction register poort A. Geeft aan of de poort op lezen of schrijven staat. (0=lezen, 1=schrijven)
$BFE301:Data direction register poort B. Deze is bedoelt voor de Centronix interface. Zonder toepassingen kunt u deze vrij programmeren.
$BFE401:TimerLO-A, timer A is in gebruik voor communicatie met het toetsenbord.
$BFE501:TimerHI-A, timer A
$BFE601:TimerLO-B, het besturingssysteem gebruikt deze voor meerdere doeleinden.
$BFE701:TimerHI-B, timer B
$BFE801:EventCounter bits0-7. Gebruikt voor het tellen van de 50Hz impulsen van de netspanning.
$BFE901:EventCounter bits8-15.
$BFEA01:EventCounter bits16-23.
$BFEB01:geen functie
$BFEC01:Serial Port. Via deze seriële poort komen de toetsenbord gegevens binnen.
$BFED01:Interrupt Control Register, hierin wordt bepaalt welke interruptbronnen een interrupt afgeven.
bit7 :**lezen:** IR, geeft aan dat er een interrupt geweest is. **schrijven:** S/C, Set/Clear, geeft aan of een bit, wat u in het ICR set of reset, door een interrupt of 0 (reset) of 1 (set) wordt gezet.
bit5,6:0
bit4 :FLAG, gezet als er een negatieve flank op de FLAG-lijn van CIA-A is.
bit3 :SP, het schuifregister van de seriële poort is vol of leeg, afhankelijk van in- of uitvoer.
bit2 :ALARM, de EventCounter en de ALARM waarde zijn aan elkaar gelijk.
bit1 :TB, onderloop van Timer B
bit0 :TA, onderloop van Timer A
$BFEE01:Stuurregister A, hierin worden de eigenschappen van timer A bepaalt.
bit7 :niet gebruikt
bit6 :SPMODE, seriële poort, 0=ingang, 1=uitgang
bit5 :INMODE, welke bron te gebruiken, 0=klok, 1=CNT-pulsen.
bit4 :LOAD, 1=geforceerd laden (strobe).
bit3 :RUNMODE, 0=continue lopen, 1=één keer (one shot)

bit2 :OUTMODE, signaal op PB6, 0= pulse, 1= toggle
bit1 :PBon, timer signaal op PB, 0=nee, 1= ja
bit0 :START, 1=start timer
$BFEF01:Stuurregister B, hierin worden de eigenschappen van timer B bepaalt.
bit7 :ALARM, 0=Time Of Day klok (TOD), 1=alarm van Event Counter
bit5,6:INMODE, 00=klok, 01=CNT-pulsen, 10=timerA, 11= timerA+CNT
bit4 :LOAD, 1=geforceerd laden (strobe)
bit3 :RUNMODE, 0= continue, 1=one-shot
bit2 :OUTMODE, signaal op PB7, 0=pulse, 1= toggle
bit1 :PBon, timer signaal op PB7, 0=nee, 1=ja
bit0 :START, 1= start timer

CIA B:
$BFD000:Poort A, seriële poort en Centronix interface.
bit7 :DTR, Data Terminal Ready signaal van seriële interface
bit6 :RTS, Ready To Send signaal
bit5 :DCD, Data Carrier Detect signaal
bit4 :CTS, Clear To Send signaal
bit3 :DSR, Data Set Ready signaal
bit2 :SEL, select signaal van de Centronix interface
bit1 :POUT, Paper Out signaal van de Centronix interface
bit0 :BUSY, Busy signaal van de Centronix interface
$BFD101:Poort B, gereserveerd voor de diskdrive's
bit7 : MTR, motor signaal voor diskdrive(s), 1=aan, 0=uit
bit6 :SEL3, selecteer drive 3
bit5 :SEL2, selecteer drive 2
bit4 :SEL1, selecteer drive 1
bit3 :SEL0, selecteer interne drive
bit2 :SIDE, welke zijde van de disk moet gelezen worden?
bit1 :DIR, richting-signaal voor de diskdrive (DIRection)
bit0 :STEP, step-signaal voor diskdrive
$BFD201:Data Direction Register A, 1= schrijven (signaal gaat naar buiten), 0=lezen
$BFD301:Data Direction Register B, idem als boven, nu voor Poort B
$BFD401:TimerLO-A, lowbyte van timer A. Timer A alleen gebruikt voor het 'timen' van de gegevens van de RS232 poort
$BFD501:TimerHI-A, hibyte van timer A
$BFD601:TimerLO-B, deze synchroniseert de Blitter met het scherm
$BFD701:TimerHI-B, hibyte van timer B
$BFD801:Event Counter bits 0-7, de Event Counter telt de impulsen van de horizontale synchronisatie van het beeldscherm



$BFD901:Event Counter bits 8-15
$BFDA01:Event Counter bits 15-24
$BFDB01:niet in gebruik
$BFDC01:Seriële Poort, niet in gebruik
$BFDD01:Interrupt Control Register. De bits zijn hetzelfde als in CIA A
$BFDE01:Stuurregister A, zie CIA A voor uitleg
$BFDF01:Stuurregister B, idem

Dit waren globaal de gegevens van de seriële poort. Hopelijk kunt u hiermee het probleem waar u nu al dagen mee bezig was, oplossen. Natuurlijk zijn er nog veel meer dingen over de CIA's bekend. Hiervoor zijn in de boekhandel verschillende goede boeken te krijgen.



AutoRequesters

Hoe vaak is het in een programma niet nodig, de simpele JA/NEE antwoorden. U kunt in C dit natuurlijk oplossen door met behulp van getchar() of scanf() invoer vanaf het toetsenbord te lezen. Toch is dit niet de echte 'Amiga Methode'. We willen juist met de muis het JA- of NEE antwoord binnenkrijgen. Het besturingssysteem geeft hiervoor een simpele oplossing. De AutoRequester's. Hoe ziet een AutoRequester er uit?
Antwoord=AutoRequest(Window,BodyText,PositiveText,NegativeText,PositiveFlags,NegativeFlags,Width,Height)

Window	pointer naar een Window-structuur
BodyText	IntuiText structuur voor de 'vraag'
Positive	TextIntuiText structuur voor het JA antwoord
NegativeText	IntuiText structuur voor het NEE antwoord
PositiveFlags	IDCMP-flag voor het JA veld
NegativeFlags	IDCMP-flag voor het NEE veld
Width	breedte van de AutoRequester
Height	hoogte van de AutoRequester

Het is dus noodzakelijk twee IntuiText structuren op te zetten. Een IntuiText structuur is de Intuition voorziening voor tekst op het beeldscherm. In de IntuiText structuur kunt u meerdere gegevens van het af te printen stuk tekst opgegeven, als welke kleur, wat voor font, waar op het beeldscherm, etcetera.
Om nu het bovenstaande verhaal te illustreren hebben we even een voorbeeld bijgevoegd.

```
#include "functions.h"
#include "exec/types.h"
#include "intuition/intuitionbase.h"

struct IntuitionBase *IntuitionBase;
struct IntuiText Einde = {
     3,0,JAM1,25,10,NULL,
     (UBYTE *)"Programma beeindigen?",
     NULL };
struct IntuiText JA = {
     3,0,JAM1,5,3,NULL,
     (UBYTE *)" Ja ",NULL };
struct IntuiText NEE = {
     3,0,JAM1,5,3,NULL,
     (UBYTE *)"NEE!",NULL };

main()
{
BOOL Antwoord;

if(!(IntuitionBase=(struct IntuitionBase     *)OpenLibrary("intuition.library",0)))
{
printf("Geen intuition.library\n");
exit(FALSE);
}
while((Antwoord=AutoRequest(IntuitionBase->ActiveWindow,&Einde,&JA,&NEE,0,0,250,60))==FALSE);

CloseLibrary(IntuitionBase);
}
```

In dit voorbeeld is te zien dat het programma in een 'while' lus blijft hangen totdat het JA gadget in de AutoRequester wordt aangeklikt. Deze actie levert een waarde TRUE op waarna uit de while lus wordt gegaan. Voor simpele JA/NEE spelletjes, etcetera is het de ideale oplossing!

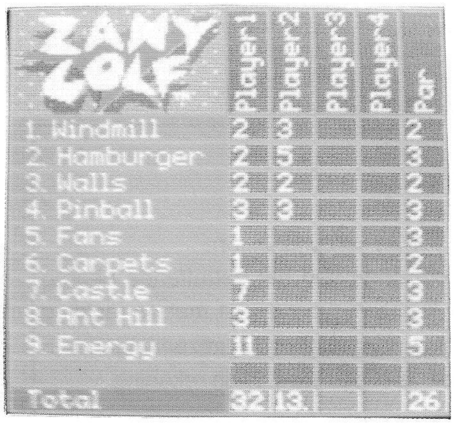
Tot besluit

Deze keer niet zoveel variatie. De ruimte voor deze hoek was snel vol door het overzicht van de Guru meditations. De volgende keer weer wat meer verschillende onderwerpen. De volgende keer dan ook voor de eerste keer onderwerpen door de lezers aangedragen. Dus blijf erbij en stuur al uw suggesties, reacties, vragen, programma's op naar ons dan ontstaat er een echte lezershoek!

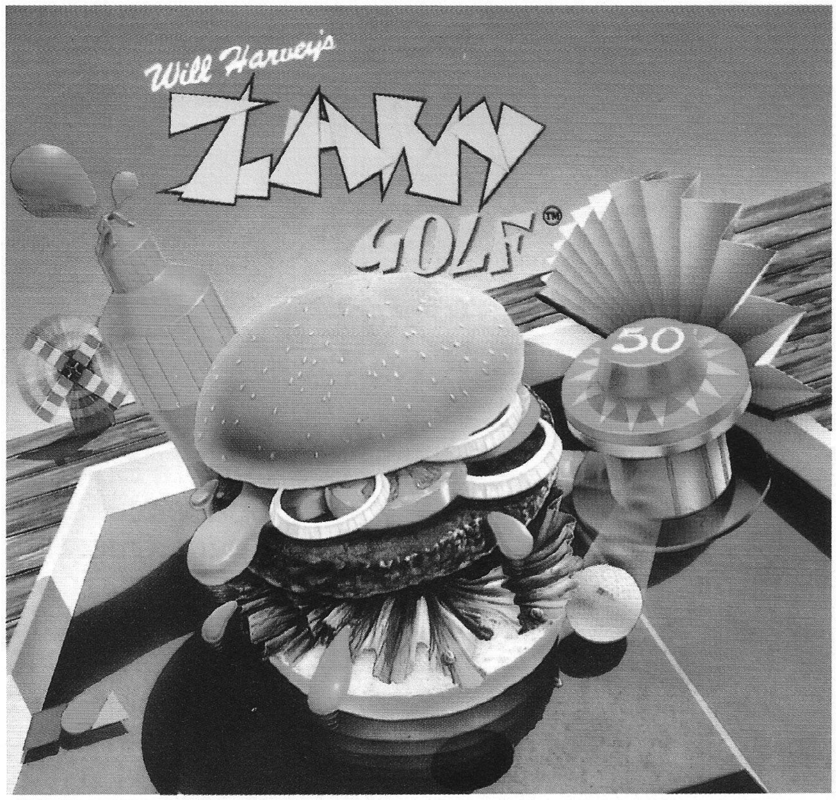
Midgetgolf is een spel dat velen aantrekt, maar helaas in Nederland niet altijd gespeeld kan worden. Het weer gooit nog wel eens roet in het eten. Electronic Arts heeft hier wat opgevonden. Een Amiga, het pakket Zany-golf en wat geduld; zie hier de ingrediënten voor vele uurtjes plezier.

Zany golf

Het spel kan met maximaal vier personen worden gespeeld en heeft negen verschillende banen. Oplopend in moeilijkheidsgraad. Bij het golven is het een goede gewoonte om te spreken over holes, waarbij wij ons dan maar aansluiten. Het is de bedoeling de bal in zo min mogelijk slagen in het putje aan het einde van de hole te slaan. Nu is het niet op alle banen mogelijk dit in één keer te doen. Van elke baan is bekend hoeveel slagen er gemiddeld over een hole worden gespeeld. Dit aantal slagen wordt "par" genoemd. Bijvoorbeeld: op baan 1 is par 3. Speelt men nu de bal in drie slagen in het putje, dan speelt men gemiddeld (wat erg goed is). Dit gemiddelde wordt bij elke baan links boven op het beeldscherm weergegeven. Het spel is voor een speler afgelopen, aan het einde van het parcours, maar ook als het maximale aantal slagen is bereikt. Rechtsboven staat het aantal slagen dat men nog over heeft om het spel te vervolgen. Bij elke baan wordt dit aantal verhoogt met het maximale aantal slagen dat er voor die baan beschikbaar is. Dus hoe meer men onder par speelt des te meer ballen blijven er voor de volgende baan over. Het spel is tijdelijk te onderbreken door het indrukken van de "P" toets. Het scoreverloop krijgt men op het beeldscherm door de "S" toets in te drukken. Het programma staat dan ook automatisch op pauze.



	Player1	Player2	Player3	Player4	Par
1. Windmill	2	3			2
2. Hamburger	2	5			3
3. Walls	2	2			2
4. Pinball	3	3			3
5. Fans	1				3
6. Carpets	1				2
7. Castle	7				3
8. Ant Hill	3				3
9. Energy	11				5
Total	32	13		1	26



Op elke baan kan het gebeuren dat er een tijdbonus wordt uitgeloofd. Het is dan zaak om de bal zo snel mogelijk in het putje te slaan. Het maximale aantal bonusslagen is vier, en dat loopt langzaam terug naar één. Een extra mogelijkheid om extra slagen te verdienen is een soort vogeltje dat soms op de baan verschijnt. Deze moet dan zo snel mogelijk geraakt worden. Het blijft opletten dus.

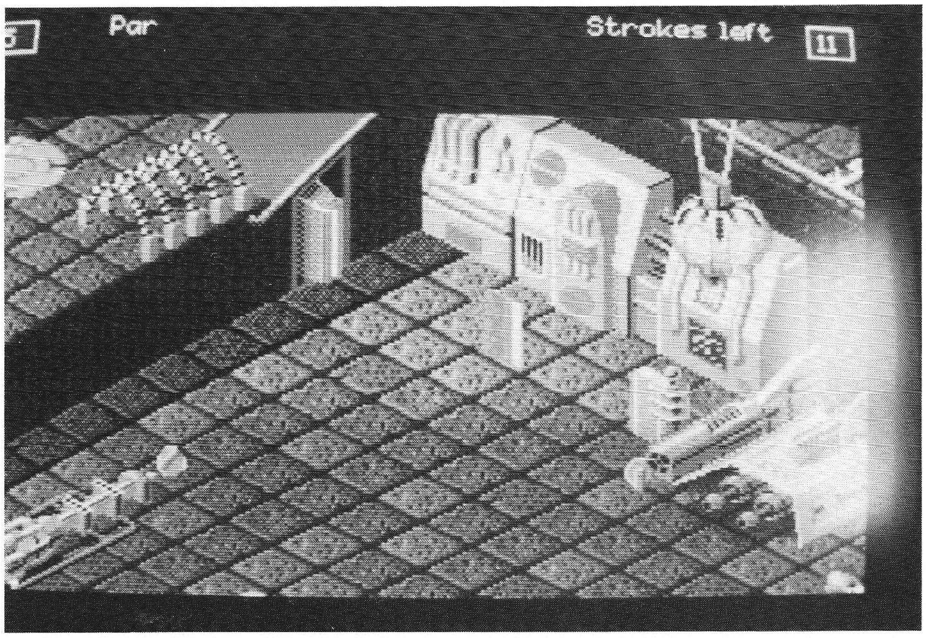
Na het opstarten van het spel moet er eerst worden ingegeven met hoeveel spelers men het spel wil spelen. Speelt men alleen dan werkt het toch veel fijner door meerdere spelers te kiezen. Is men namelijk door zijn slagen heen dan start het spel opnieuw. Nu komen we gelijk op een nadeel van het spel: Het wordt geheel opnieuw geladen. Dit neemt de nodige tijd in beslag, waarin men maar zit te wachten. Speelt men met meerdere spelers dan kan er door de andere spelers gewoon worden doorgespeeld tot de laatste speler af is. Nadat het aantal spelers is ingegeven verschijnt er een overzicht van de eerste hole. Hierop is goed te zien waar de bal moet worden afgeslagen en waar de bal heen moet. Het is zaak de baan van de bal goed te bestuderen. Het starten van deze hole gebeurt door het indrukken van de linker muisknop. Het afslaan is erg eenvoudig,

het kruisje op de bal zetten, de linkermuisknop indrukken. Nu houden we de knop ingedrukt terwijl we de muis naar achter trekken. De richting kunnen we nu bepalen door een denkbeeldige lijn te trekken, vanaf het kruisje, over de bal, naar het hole of een andere plaats waar we heen willen slaan. Hoe verder we het kruisje van de bal zetten des te harder zal de bal gaan. Om dit geheel beter te kunnen beoordelen verschijnt er een stippellijn tussen het kruisje en de bal. Denk er wel om meestal sla je veel en veel te hard.

De eerste baan is de WindMill, de windmolenbaan. Aan het eind van het eerste traject staat een windmolen hevig te draaien. Nu is het de kunst de bal in één keer door de opening in het midden te spelen. Het kan, maar het is niet makkelijk. Lukt dit, dan wordt het beloond met een extra vrije slag. Gaat dit niet doordat bijvoorbeeld net één van de wieken er voor zit, dan is er nog niets aan de hand. Bij het terugrollen loopt de bal van de helling zo naar het tweede traject. Van hieruit is het een kwestie van nauwkeurig en rustig spelen. Baan twee is de be-

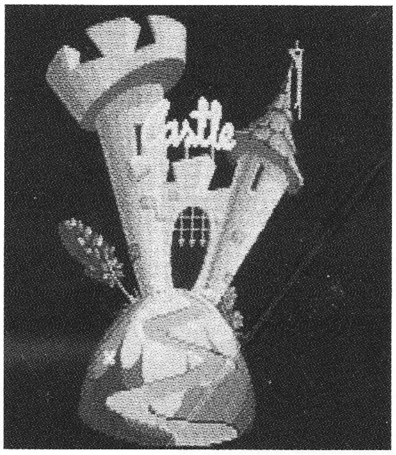


roemde hamburger baan. Hier staat een levensecht broodje hamburger, inclusief uiringen, boven op het hole. Door de muisknop een aantal keren snel achter elkaar in te drukken begint dit broodje te dansen. Nu komt de hole vrij en door rustig via de ketchup te spelen loopt de bal in het hole. Zeker hier geldt, beter te langzaam dan te snel, want omhoog spelen is erg moeilijk. De muren, baan drie, hoeft helemaal geen problemen op te leveren: even goed kijken en nadenken. Baan vier, de flipperkast, is één van de spectaculaire onderdelen. Gebruik de flippers op het juiste moment, alles wat kan branden moet gaan branden en daarna ook nog proberen de bal in



het gat te schieten. Valt de bal per ongeluk achter één van de blokjes, dan even opletten. Omdat de bal nu niet te zien is en het blokje op de bal geplaatst moet worden, kan het spel, zoals mij wel eens gebeurt is, hier afgelopen zijn. Gelukkig staat bij het starten het kruisje precies op de bal, nu niet schuiven met de muis maar direct de muisknop indrukken en dan pas verplaatsen. Nu is het niet moeilijk meer om de bal in het putje te spelen.

Ventilatoren spelen een grote rol bij de volgende baan. Het is zaak deze voor je te laten werken. Door met de muis te draaien zijn de ventilatoren te draaien. Zij kunnen daardoor de bal een mooie beweging voorwaarts geven. Maar denk erom, selectief gebruiken, anders gaan ze tegen je werken. Een prachtige baan is het Magic Carpet, en magie komt hier zeker aan te pas. Is de bal op één van de geblokte velden, let er dan op dat je de muis niet stil houd. Door deze in de juiste richting te bewegen kan de bal in de richting van de hole gebracht worden, op magische wijze. Hier blijkt het vaak mogelijk de bal in éénmaal in het hole te spelen. Eén van de mooiste grafische afbeeldingen is ongetwijfeld het Kasteel. Hier is alles veel moeilijker dan het lijkt. Rechtdoor spelen, want alleen de snelheid is belangrijk. Een extra vrije bal krijg je, als de bal in het midden door de kasteelpoort wordt gespeeld. Op de binnenplaats moet voorzichtig worden gespeeld, de muurtjes blijken grote obstakels te zijn. Het hole in de volgende baan ligt boven op een heuvel. Dus te hard, of te zacht wordt onmiddellijk afgestraft. Om het geheel nog wat moeilijker te maken verplaatst het hole ook nog van plaats. De bumpers aan het eind van de heuvel kunnen de bal weer wat extra snelheid geven. Het laatste hole is ook het moeilijkste. Voor elke baan wordt bij het overzichtsscherm een tip gegeven. Dit gebeurt bij deze hole niet. Met andere woorden: zoek het zelf maar uit. Wij zullen U alleen vertellen dat de knoppen op de computer een belangrijke rol spelen bij het doorlopen van deze baan.



Elk onderdeel wordt begeleid door een muziekje in stereo. Doet men te lang over een baan, dan wordt dit soms wel wat eentonig, maar het geluid van de in het hole vallende balletje maakt veel goed. Jammer dat er maar negen holes zijn, maar Electronic Arts kennende zal het wel niet al te lang duren voor er diskettes met extra holes op de markt komen.

Soft Wir-War

Bert Tiev & Wouter Meijer



Dat de Amiga zich uitstekend leent voor het maken en vervormen van geluiden is bekend. In toenemende mate maken amateurs en professionals gebruik van de muzikale kwaliteiten van dit medium. Een spectrum-analyser kan in dit verband prima diensten bewijzen als 'grafisch hulpje'. Vandaar dat we ditmaal in onze zelfbouw-rubriek het bouwen van zo'n analyser behandelen en de sampler, die hierbij nodig is.

Amiga Zelfbouw

sampler & spectrum-analyser

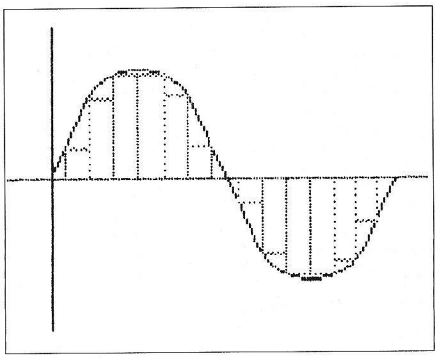
Een spectrum-analyser geeft geluiden op een grafische manier weer. Dit kan veel voordelen hebben bij het zelf maken of het vergelijken van bepaalde geluiden via de computer. Bij de Amiga is daar een stukje hardware voor nodig en een simpel programmaatje. In dit artikel geven we een voorbeeld van de software en laten zien hoe je de benodigde hardware zelf bouwt.



De Spectrum Analyser aan het werk

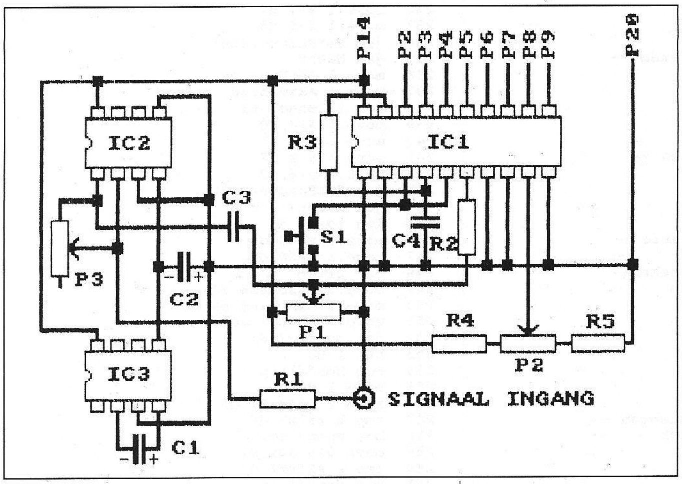
Bouw uw eigen spectrum-analyser

De Amiga wekt het geluid op een andere manier op, dan de meeste andere computers in die prijsklasse (en nog duurdere).
De Amiga wekt het geluid digitaal op, in tregenstelling tot analoog d.m.v. een geluids-chip. De Amiga heeft ook een geluids-chip, maar deze werkt digitaal en heet Paula. Paula doet nog meer dan het geluid opwekken, namelijk de seriële poort regelen, interrupt afhandelen, en analoge joysticks en de diskdrive besturen.
Er wordt gezegd dat de Amiga compact-disc kwaliteit heeft, maar dat is absoluut niet waar. Bij de Amiga zijn de geluidsregisters 8 bits breed. De data kunnen dus maximaal een waarde van 255 aannemen (dit is 2^8). Dit is te zien in afbeelding 1, (de sinus functie). Een CD-speler werkt met een samplebreedte van 16 bits. Deze kan dus de maximale waarde van 65535 aannemen (dit is 2^16). Een compact-disc speler heeft dus een oplossing die 256 keer zo hoog is. Als je goed luistert naar het geluid van de Amiga hoor je ook iets ruis.



Afbeelding 1

Voor de kwaliteit van het geluid is eveneens van belang dat de sample-frequentie voor een natuurgetrouwe sample (een sample is een blokje met data, dit zijn waarden die alle tezamen het geluid vormen) minstens 2 maal zo hoog moet zijn als de hoogste frequentie in het te samplen signaal. Het menselijke oor kan geluiden tot bijna 20 kHz waarnemen. Voor een goede sample heb je dus een sample-frequentie nodig van minstens 40 kHz. Een compact--disc speler heeft een sample-frequentie van 44,1 kHz. De Amiga verwekt het geluid met een maximale sample-frequentie van 28,2 kHz. Behalve bij de nieuwe Audiomaster 2 daar staat dat deze een sample-frequentie heeft van 44,1 kHz. Dit kan in principe niet omdat de hardware dit nooit aankan.
Vijf seconde muziek met een sample-frequentie van 44,1 kHz (zoals bij een compact-disc speler) neemt een geheugen in van ruim 440 kilobytes (dit is mono).
Vijf seconde muziek op een Amiga in de best mogelijke kwaliteit neemt een geheugen in van 140 kilo bytes.



Afbeelding 2: Het schema

Het ledje

Het ledje kan op de Amiga 500/2000b worden uit of aan worden geschakeld. Als het ledje wordt uitgeschakeld wordt ook het geluids-filter uitgeschakeld. Als het ledje aan staat worden alle frequenties boven de 7 kHz afgebogen naar beneden. Als deze uitstaat zal dit pas gebeuren bij 14 kHz. Het ledje kan vanuit BASIC als volgt worden geschakeld:

```
POKE 12575233,3
POKE 12574721,0        powerled aan
POKE 12574721,2        powerled uit
```

Vanuit assembler gaat het als volgt:

```
move.b #3,12575233
move.b #0,12574721     powerled aan
move.b #2,12574721     powerled uit
```

De hardware

De schakeling bestaat uit twee gedeeltes, het ene gedeelte is de eigenlijke AD omzetter en de andere helft is een eenvoudige versterker schakeling. De AD omzetter zet het signaal om van analoog naar digitaal (8 bits) zodat de Amiga het kan verwerken. De versterker versterkt het binnenkomende audiosignaal. De potmeter P1 dient er voor om de sampler grof af te regelen, zodat er een rechte streep in het midden van het scherm staat als er geen signaal op de sampler aanwezig is. De potmeter genaamd P2 dient ervoor om de streep fijn af te regelen. Voor dat er ook maar iets is af te regelen moet eerst de drukknop S1 worden ingedrukt om de AD omzetter te starten.

De potmeter P3 is er om het volume van 0 tot maximaal te regelen, dit is ook op het scherm te zien als er signaal op staat. Je kunt ook gewoon de versterker harder zetten.

De bouw

Het schema is klein genoeg om op een experimenteer-printje te bouwen. Het is geschikt voor iedereen die weet wat een soldeerbout is. De bouwkosten bedragen ongeveer *f* 35,-.
De 'sampler' wordt op de parallel poort aangesloten zoals in het schema staat afgebeeld (afbeelding 2). De nummers met een P ervoor komen overeen met de nummers van de parallele connector bij de Amiga 500. Denk erom de condensators en IC's goed om te zetten.

Benodigde onderdelen:

- IC1 ADC0804
- IC2 MC1458
- IC3 ICL7660
- P1 Potmeter 10 K Ohm
- P2 Potmeter 1 K Ohm
- P3 Potmeter 100 K Ohm
- C1,C2 10 micro Frad/63 V
- C3 680 nano Farad
- C4 150 pico Farad
- R1,R2 1 K Ohm
- R3,R4,R5 10 K Ohm
- S1 drukknop
- 1 experimenteer print (70X80 mm)
- 1 25 polige sub D male connector (voor A500/A2000)
- Minimaal 10 aderig draad (ongeveer 40 cm lang)

De software

De software dient ervoor om de waarden die de computer binnen krijgt via de sampler om te zetten in grafieken. Het programma is menu gestuurd. Als het programma is ingetikt en de sampler gebouwd dient de sampler te worden afgeregeld. Dit doet men als volgt: De sampler wordt op de computer aangesloten als deze uitstaat, vervolgens wordt de software ingeladen. Nu wordt menu keuze Graph4 gekozen via de rechter muistoets. Er verschijnt dan een streep op het scherm. Vervolgens moet de drukknop op de sampler worden ingedrukt. De lijn zal nu op een andere plaats verschijnen. Nu moet men met de potmeters P1 en P2 de lijn in het midden van het scherm zetten door aan de potmeters te draaien. Dit alles gebeurt terwijl er geen signaal op de sampler is aangesloten. Als de lijn in het midden staat kan men een signaal op de sampler aansluiten. Dit kan gedaan worden als de computer aanstaat.
Men kan tussen de verschillende Graph modes kiezen door eerst op de linker muistoets te drukken om de oude Graph mode te verlaten en vervolgens een nieuwe te kiezen met behulp van de rechter muistoets en de menu's boven in het scherm. Onder in het scherm staat vermeld in welke mode het programma staat. Als bonus kan men ook nog luisteren naar de muziek die op de sampler is aangesloten. De kwaliteit van deze sampler is niet zo best, maar goed genoeg voor dit programma. Dit programma is eveneens bruikbaar met de meeste andere samplers, in dat geval heeft men meer aan de monitor functie.
Als extraatje is er een roterende muispointer aanwezig tijdens het kiezen van een van de mogelijkheden.

Het intikken van de listing

De getallen voor de regels dienen niet te worden ingetikt. Deze zijn zuiver en alleen om aan te geven waar een nieuwe regel begint. De Seka assembler die wij hebben gebruikt om dit programma te maken zal zelf de getallen neerzetten. De regels met puntkomma (;) ervoor bevatten alleen maar commentaar en uitleg wat er in dat programma deel gebeurt. Zo kan men ook delen uit het programma halen en in je eigen programma gebruiken.
Het programma moet eerst worden geassembleerd (A) en vervolgens worden weggeschreven met WO (Write as Object file). Nu kan het programma vanuit de CLI worden ingeladen via de naam die bij WO is ingegeven. De disk waar het programma op staat moet via de Preferences op 80 karakters zijn ingesteld. Het programma werkt in de hoogste resolutie. Hier komt de Seka listing:

```
;*** Spectrum Analyser ***
;* (c) Hans van der Pol *
;publicatie recht:
;Commodore Info
;-------------------------
```

```
;Je hebt een eenvoudige sampler
;nodig bij dit programma om
;de muziek om te zetten naar waarden
;voor de spectrum analyser (D - A)
;--------------------------
;Uit graph1,2,3,4 door op linker muis-
;toets te drukken!!!!
;bij voorkeur op een disk zetten
;waarbij de preferences op 80
;karakters per regel staat!!!
;--------------------------
;-- Offsets --
openwindow        = -204
closewindow       = -72
openscreen        = -198
closescreen       = -66
openlib           = -408
closelibrary      = -414
setapen           = -342
setbpen           = -348
writepixel        = -324
readpixel         = -318
delay             = -198
rectfill          = -306
setmenustrip      = -264
clearmenustrip    = -54
getmsg            = -372
move              = -240
draw              = -246
waittof           = -30-240
setrgb4           = -30-258
setpointer        = -30-240
viewportadr       = -30-270
supervisor        = -30
drawborder        = -108
printitext        = -216

;-- extra adressen/waarden --
execbase          = 4
buffer            = $50000
len               = $f
speed             = 25

;-- adressen voor parallel --
;-- poort. (uitlezen) --
set               = $bfe301
lees              = $bfe101
PRA               = $bfd000
PRB               = $bfe101
DDRA              = $bfd200
DDRB              = $bfe301

;-- custom adressen --
;zie Hardware manuel of Intern
aud0lch           = $dff0a0
aud0len           = $dff0a4
aud0per           = $dff0a6
aud0vol           = $dff0a8
aud0dat           = $dff0aa
aud1lch           = $dff0b0
aud1len           = $dff0b4
aud1per           = $dff0b6
aud1vol           = $dff0b8
aud1dat           = $dff0ba

;-- custom --
DMAcon            = $dff096
color00           = $dff180

;-- eerst alle libraries openen --
start:
;-- intuition.library openen --
 move.l execbase,a6
 lea intname,a1
 jsr openlib(a6)
 move.l d0,intbase

;-- graphics.library openen --
 move.l execbase,a6
 lea gfxname,a1
 jsr openlib(a6)
 move.l d0,gfxbase

;-- dos.library openen --
 move.l execbase,a6
 lea dosname,a1
 jsr openlib(a6)
 move.l d0,dosbase

;-- screen openen --
 move.l intbase,a6
 lea screen_defs,a0
 jsr openscreen(a6)
 move.l d0,screenhd

;-- window openen --
 move.l intbase,a6
 lea window_defs,a0
 jsr openwindow(a6)
 move.l d0,windowhd

;-- eigen rastport eruit --
;-- vissen! --
 move.l windowhd,a1
 move.l 50(a1),a1
 move.l a1,rastport

;-- parallel poort zetten --
 move.b #$00,set

;-- menu strip zetten --
 move.l intbase,a6
 move.l windowhd,a0
 lea project,a1
 jsr setmenustrip(a6)

;-- message balk tekenen --
 lea bordertje,a1
 move.w #0,d0
 move.w #0,d1
 bsr border

;-- message neerzetten --
 lea text6,a1
 move.l #80,d0
 move.l #468,d1
 bsr print
;-- muis-pointer rotatie --
;-- voorbereiden        --
;-- aktuele window zoeken--
;-- + viewport zoeken    --
 move.l IntBase,a0
 move.l 52(a0),AktWin
 move.l IntBase,a6
 move.l AktWin,a0
 jsr ViewPortAdr(a6)
 move.l d0,AktVP

;-- hier worden de kleuren --
;-- voor pointer gezet      --
 move.l AktVP,a0
 move.l GfxBase,a6
 move.l #17,d0
 move.l #0,d1
 move.l #0,d2
 move.l #0,d3
;--kleur zetten --
 jsr SetRGB4(a6)
 move.l AktVP,a0
 move.l #18,d0
 move.l #13,d1
 move.l #2,d2
 move.l #2,d3
 jsr SetRGB4(a6)
 move.l AktVP,a0
 move.l #19,d0
 move.l #15,d1
 move.l #12,d2
 move.l #12,d3
 jsr SetRGB4(a6)

;-- hoofdloop, hier tast --
;-- het programma de menus --
;-- af of er wat is gekozen --
hoofdloop:
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image0,a1
;-- pos. muispointer --
 move.l #14,d0
 move.l #16,d1
 move.l #-7,d2
 move.l #-7,d3
;-- nieuwe muispointer --
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image1,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-6,d2
 move.l #-8,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image2,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-5,d2
 move.l #-7,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image3,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-4,d2
 move.l #-6,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image4,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-5,d2
 move.l #-5,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image5,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-6,d2
 move.l #-4,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image6,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-7,d2
 move.l #-5,d3
 jsr SetPointer(a6)
 jsr Wacht
 move.l IntBase,a6
 move.l AktWin,a0
 lea.l Image7,a1
 move.l #14,d0
 move.l #16,d1
 move.l #-8,d2
 move.l #-6,d3
 jsr SetPointer(a6)
 jsr Wacht
 lea text0,a1
 move.l #100,d0
 move.l #480,d1
 bsr print
 move.l execbase,a6
 move.l windowhd,a0
 move.l 86(a0),a0
 jsr getmsg(a6)
 tst.l d0
 beq hoofdloop
 move.l d0,a0
 move.l $14(a0),d6
 cmp.b #$100,d6
 bne hoofdloop
 move $18(a0),d7
 cmp.w #$ffff,d7
 beq hoofdloop
 cmp.w #$f820,d7
 beq xniks
 cmp.w #$f840,d7
 beq xsample
 cmp.w #$f801,d7
 beq xgraph1
 cmp.w #$f821,d7
 beq xgraph2
 cmp.w #$f841,d7
 beq xgraph3
 cmp.w #$f861,d7
 beq xgraph4
 cmp.w #$f800,d7
 bne hoofdloop

;-- nu wordt alles afgesloten --
stop:
;-- menus weghalen --
 move.l IntBase,a6
 move.l windowhd,a0
 jsr clearmenustrip(a6)

;-- window sluiten --
 move.l intbase,a6
 move.l windowhd,a0
 jsr closewindow(a6)

;-- screen sluiten --
 move.l intbase,a6
 move.l screenhd,a0
 jsr closescreen(a6)

;-- intuition.library sluiten --
 move.l execbase,a6
 move.l intbase,a1
 jsr closelibrary(a6)

;-- graphics.library sluiten --
 move.l execbase,a6
 move.l gfxbase,a1
 jsr closelibrary(a6)

;-- dos.library sluiten --
 move.l execbase,a6
 move.l dosbase,a1
 jsr closelibrary(a6)

;-- terug naar CLI alsof er --
;-- niks is gebeurt --
 rts

;-- tekst afdrukken --
print:
 move.l intbase,a6
 move.l rastport,a0
 jsr printitext(a6)
 rts

;-- border (lijnen) tekenen --
border:
 move.l intbase,a6
 move.l rastport,a0
 jsr drawborder(a6)
 rts

;-- wacht voor nieuwe --
;-- muis pointer --
Wacht:
 move.l DosBase,a6
 move.l #2,d1
 jsr Delay(a6)
 move.l GfxBase,a6
 jsr WaitTOF(a6)
 move.l GfxBase,a6
 jsr WaitTOF(a6)
 rts

;-- menu keuzen graph1 --
; -- punten zetten --
xgraph1:
;-- pen zetten --
;-- en tekst neerzetten --
 lea text2,a1
```

```
 move.l #100,d0
 move.l #480,d1
 bsr print

 move.l rastport,a1
 move.l gfxbase,a6
 move.l #1,d0
 jsr setapen(a6)

;-- waarde uitlezen van --
;-- parallel poort en    --
;-- omzetten in y-coord.--
;-- daarna punt zetten  --
xgraph1loop1:
 move.l rastport,a1
 move.l gfxbase,a6
 move.l tel0,d0
 move.b lees,d2
 move.w d2,d1
 and.w #%11111111,d1
 move.b d1,d1
 sub.b #5,d1
 jsr writepixel(a6)

;-- zijn we al aan het eind --
;-- van het scherm ?        --
 add.l #1,tel0
 cmp.l #600,tel0
 beq funny
 btst #6,$bfe001
 bne xgraph1loop1
;-- terug naar hoofdloop --
;-- en tekst weghalen    --
 lea text0,a1
 move.l #100,d0
 move.l #480,d1
 bsr print

 bra hoofdloop

;-- graph2 --
;-- deze trekt lijnen tussen twee --
;-- binnengekomen punten (verikaal) --
xgraph2:
;-- pen kleur zeten --
;-- en tekst neerzetten --
 lea text3,a1
 move.l #100,d0
 move.l #480,d1
 bsr print

 move.l rastport,a1
 move.l gfxbase,a6
 move.l #1,d0
 jsr setapen(a6)

;-- cursor op goede plaats zeten --
;-- links op het scherm --
 move.l gfxbase,a6
 move.l rastport,a1
 move.w #20,d0
 move.w #255,d1
 jsr move(a6)

;-- cursur naar nieuwe pos. schuiven --
;-- y-coord. constant opschuiven --
xgraph2loop1:
 move.l gfxbase,a6
 move.l rastport,a1
 move.w #255,d1
 move.w tel4,d0
 jsr move(a6)

;-- nieuwe coord. van --
;-- parallel poort uitlezen --
;-- om een lijn te trekken tussen --
;-- oude y-coord. en nieuwe --
;-- de x-coord. blijft gelijk --
 move.l gfxbase,a6
 move.l rastport,a1
 move.b lees,d1
;-- masker erover heen --
 and.w #$0000000011111111,d1
 add.w #127,d1
 move.w d1,tel5
 move.w tel4,d0
 jsr draw(a6)

;-- cursor naar nieuwe coord. --
;-- schuiven (wordt constant door --
;-- getelt) -
 move.l gfxbase,a6
 move.l rastport,a1
 move.w tel5,d1
 move.w tel4,d0
 jsr move(a6)

;-- muistoets? --
 btst #6,$bfe001
 beq hoofdloop

;-- alle coord. bijwerken   --
;-- en vergelijken of het   --
;-- scherm al moet worden   --
;-- schoongemaakt.          --
;-- als tel4 wordt verhoogd --
;-- gaat de grafiek sneller --
;-- 600 gedeelt door tel4   --
;-- moet wel een geheel     --
;-- opleveren!!             --
 add.w #5,tel4 ;verhogen is versnellen
 cmp.w #600,tel4
 bne xgraph2loop1
funny2:
;-- pen ontzichtbaar maken --
 move.l rastport,a1
 move.l gfxbase,a6
 move.l #0,d0
 jsr setapen(a6)

;-- vierkant over oude  --
;-- tekening heen       --
 move.l gfxbase,a6
 move.l rastport,a1
 move.l #3,d0
 move.l #9,d1
 move.l #620,d2
 move.l #450,d3
 jsr rectfill(a6)

;-- tel4 bijstellen --
 move.w #20,tel4
 bra xgraph2

;-- graph3, balken laten    --
;-- stuiteren op coord. van --
;-- parallel poort          --
xgraph3:
;-- nieuwe copperlist --
 lea text4,a1
 move.l #100,d0
 move.l #480,d1
 bsr print
 move.l gfxbase,a0
 add.l #$32,a0
 move.w #$0080,$dff096
 move.l (a0),oude_copper
 move.l #nieuwe_copper,(a0)
 move.w #$8080,$dff096
 move.l #0,tel5

xgraph3loop2:
;-- parallel poort uitlezen --
 move.b lees,tel5
 move.b tel5,d1

xgraph3loop3:
 sub.w #$64,cop1
 sub.w #$64,cop2

;-- balk omhoog tot ie   --
;-- de waarde tel5       --
;-- heeft bereikt (par.) --
 move.l dosbase,a6
 move.l #0,d1
 jsr delay(a6)
 add.b #1,tel5
 cmp.b #255,tel5
 bne xgraph3loop3

;-- standaard waarde terug --
 move.w #$e50f,cop1
 move.w #$ef0f,cop2

;-- muistoets? --
 btst #6,$bfe001
 bne xgraph3

;-- tekst weghalen --
 lea text0,a1
 move.l #100,d0
 move.l #480,d1
 bsr print

;-- gewone copperlist terug --
xstop:
 move.l gfxbase,a0
 add.l #$32,a0
 move.w #$0080,$dff096
 move.l oude_copper,(a0)
 move.w #$8080,$dff096

 bra hoofdloop

;-- graph4, scope scherm    --
;-- deze verbind lijnen met --
;-- elkaar zodat er een     --
;-- scope scherm ontstaat   --
xgraph4:
;-- penkleur zetten --
;-- tekst neerzetten --
 lea text5,a1
 move.l #100,d0
 move.l #480,d1
 bsr print

 move.l rastport,a1
 move.l gfxbase,a6
 move.l #1,d0
 jsr setapen(a6)

;-- cursor op begin pos. --
 move.l gfxbase,a6
 move.l rastport,a1
 move.w #20,d0
 move.w #255,d1
 jsr move(a6)

xgraph4loop1:
;-- parallel poort uitlezen --
;-- byte omzetten in word   --
 move.l gfxbase,a6
 move.l rastport,a1
 move.b lees,d1
 and.w #%0000000011111111,d1
 add.w #127,d1
 move.w d1,tel5
 move.w tel4,d0
;-- lijn trekken naar nieuwe coord. --
 jsr draw(a6)

;-- cursor opnieuw neerzetten --
 move.l gfxbase,a6
 move.l rastport,a1
 move.w tel5,d1
 move.w tel4,d0
 jsr move(a6)

;-- muistoets? --
 btst #6,$bfe001
 beq hoofdloop

;-- tel snelheid staat hier        --
;-- deze kan worden verhoogd       --
;-- zodat het knipperen minder     --
;-- wordt. Maar de grafiek wordt --
;-- wel blokkeriger!!! 630/tel4  --
;-- moet wel een geheel getal zijn-
 add.w #5,tel4  ;verhogen is versnelen
 cmp.w #630,tel4
 bne xgraph4loop1

;-- scherm schoonmaken m.b.v.   --
;-- vierkant in achtergrond kleur--
funny4:
;-- kleur zetten --
 move.l rastport,a1
 move.l gfxbase,a6
 move.l #0,d0
 jsr setapen(a6)

;-- vierkant --
 move.l gfxbase,a6
 move.l rastport,a1
 move.l #3,d0
 move.l #9,d1
 move.l #630,d2
 move.l #450,d3
 jsr rectfill(a6)

;-- tel4 bijwerken en terug --
 move.w #20,tel4
 bra xgraph4
 bra hoofdloop

;-- hier gebeurt niets --
;-- hier kan men zelf  --
;-- iets aan toevoegen --
xniks:
 bra hoofdloop

;-- geluid weergeven            --
;-- de bij dit artikel horende --
;-- sampler is alleen bedoeld  --
;-- als spectrum analyser.     --
;-- het geluid dat je hier     --
;-- hoord is waardeloos en     --
;-- alleen bruikbaar bij betere--
;-- samplers (maar wel handig) --
xsample:
;-- tekst neerzetten --
 lea text1,a1
 move.l #100,d0
 move.l #480,d1
 bsr print
 move.l #buffer,a0
 move.l #len,d0

xsampleclear:
 move.b #0,(a0)+
 dbra d0,xsampleclear

;-- processor in supervisor mode --
 move.l execbase,a6
 lea super,a5
 jsr supervisor(a6)

super:
 addq.l #8,a7
 move #$2700,sr
;- parallel poort zetten --
 move.b #%00000000,DDRB
;-- initialiseren --
 move.b DDRA,d0
 andi.b #%11111010,d0
 ori.b #%00000100,d0
 move.b d0,DDRA

;-- audio kanaal data 0   --
 move.l #buffer,aud0lch
;-- audio kanaal data 1   --
 move.l #buffer,aud1lch
;-- volume kanaal 0 voluit --
 move.l #64,aud0vol
;-- volume kanaal 1 voluit --
 move.l #64,aud1vol
;-- rate kanaal 0 zetten   --
 move.w #speed,aud0per
;-- rate kanaal 1 zetten   --
 move.w #speed,aud1per
;-- lengte data kanaal 0   --
 move.w #len/2,aud0len
;-- lengte data kanaal 1   --
 move.w #len/2,aud1len
;-- audio dma aanzetten    --
 move #$8003,dmacon

xsampleloop:
 lea buffer,a0
```

```
 move.w #len,d3
mus:
 ori.b #%00000100,PRA
 andi.b #%11111011,PRA
ooievaar:
 andi.b #%00000100,PRA
 beq.s ooievaar
 clr.w d1
 move.b PRB,d1
 eori.b #128,d1
 move.b d1,(a0)+
;-- geeft aan dat er data binnenkomt --
;-- kleiner dan 200 dan rood          --
 cmp.b #200,d1
 bpl poesje  ;kleiner dan 200 dan rood
 move.w #$0f00,color00
poesje:
 dbra d3,mus
 btst #6,$bfe001
 bne xsampleloop
;-- dma uit
 move.w #$0003,DMAcon
;-- processor in user mode --
 move #0,sr
;-- tekst weghalen --
 lea text0,a1
 move.l #100,d0
 move.l #480,d1
 bsr print
 bra hoofdloop
funny:
 move.l rastport,a1
 move.l gfxbase,a6
 move.l #0,d0
 jsr setapen(a6)
 move.l gfxbase,a6
 move.l rastport,a1
 move.l #1,d0
 move.l #7,d1
 move.l #620,d2
 move.l #290,d3
 jsr rectfill(a6)
 move.l #20,tel0
 bra xgraph1
even
oude_copper:
 dc.l 0
;-- copperlist voor graph3 --
even
nieuwe_copper:
 dc.w $0180,$0fff
cop1:
 dc.w $e50f,$fffe
 dc.w $0180,$0000
cop2:
 dc.w $ef0f,$fffe
 dc.w $0180,$0fff
 dc.w $ef0f,$fffe
 dc.w $ffff,$fffe
;-- screen struktuur --
even
screen_defs:
 dc.w 0,0,640,512,3
 dc.b 0,1
 dc.w $8004,15
 dc.l 0,title,0,0
titel:
 dc.b 'Spektrum analyser (c) Hans van der Pol',0
;-- window struktuur --
even
window_defs:
 dc.w 0,9,640,495
 dc.b 0,2
 dc.l $100,%0000000000000100000000000,0,0,windownaam
screenhd: dc.l 0
 dc.l 0
 dc.w 150,50,640,245,15
even
windownaam:
 dc.b 'Danny Brandt ontwikkelde de hardware',0
;-- menu strukturen --
even
Project:
 dc.l command
 dc.w 20,0,80,10,1
 dc.l menutext1
 dc.l quit
 dc.w 0,0,0,0
even
command:
 dc.l 0
 dc.w 101,30,75,10,1
 dc.l menutext2
 dc.l graph1
 dc.w 0,0,0,0
;-- onder menu strukturen --
even
quit:
 dc.l clear
 dc.w 0,0,124,10,%10011010
 dc.l 0
 dc.l quittext
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
clear:
 dc.l sample
 dc.w 0,10,124,10,%10011010
 dc.l 0
 dc.l cleartext
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
sample:
 dc.l 0
 dc.w 0,20,124,10,%10011010
 dc.l 0
 dc.l sampletext
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
graph1:
 dc.l graph2
 dc.w 0,0,124,10,%10011010
 dc.l 0
 dc.l graph1text
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
graph2:
 dc.l graph3
 dc.w 0,10,124,10,%10011010
 dc.l 0
 dc.l graph2text
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
graph3:
 dc.l graph4
 dc.w 0,20,124,10,%10011010
 dc.l 0
 dc.l graph3text
 dc.l 0
 dc.b 0
even
 dc.l 0,0
even
graph4:
 dc.l 0
 dc.w 0,30,124,10,%10011010
 dc.l 0
 dc.l graph4text
 dc.l 0
 dc.b 0
even
 dc.l 0,0
;-- tekst strukturen --
even
quittext:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l quittext2
 dc.l 0
even
cleartext:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l cleartext2
 dc.l 0
even
sampletext:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l sampletext2
 dc.l 0
even
graph1text:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l graph1text2
 dc.l 0
even
graph2text:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l graph2text2
 dc.l 0
even
graph3text:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l graph3text2
 dc.l 0
even
graph4text:
 dc.b 0,1,0
even
 dc.w 3,1
 dc.l 0
 dc.l graph4text2
 dc.l 0
;-- tekst --
even
menutext1:
 dc.b 'Project',0
menutext2:
 dc.b 'Graphic',0
quittext2:
 dc.b ' Quit',0
cleartext2:
 dc.b ' niks',0
sampletext2:
 dc.b ' Monitor',0
graph1text2:
 dc.b ' Graph 1',0
graph2text2:
 dc.b ' Graph 2',0
graph3text2:
 dc.b ' Graph 3',0
graph4text2:
 dc.b ' Graph 4',0
;-- data --
even
windowhd:
 dc.l 0
intbase:
 dc.l 0
gfxbase:
 dc.l 0
dosbase:
 dc.l 0
rastport:
 dc.l 0
tel0:
 dc.l 20
tel1:
 dc.l 125
tel3:
 dc.l 0
tel4:
 dc.l 20
tel5:
 dc.l 0
intname:
 dc.b 'intuition.library',0
gfxname:
 dc.b 'graphic.library',0
dosname:
 dc.b 'dos.library',0
even
text0:
 dc.b 0,1
 dc.b 4
even
 dc.w 0,0
 dc.l 4
 dc.l txt0
 dc.l 0
```

Amiga

```

txt0:               dc.b
',0

even
text1:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l txt1
 dc.l 0

txt1: dc.b 'monitoring',0

even
text2:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l txt2
 dc.l 0

txt2: dc.b 'writing pixels',0

even
text3:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l txt3
 dc.l 0

txt3: dc.b 'Graphic analyser activated',0

even
text4:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l txt4
 dc.l 0

txt4: dc.b 'New copperlist activated',0

even
text5:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l text5
 dc.l 0

txt5: dc.b 'Scope activated',0

even
text6:
 dc.b 1,0
 dc.b 0
even
 dc.w 0,0
 dc.l 0
 dc.l txt6
 dc.l 0

txt6: dc.b 'message:',0

even
bordertje:
 dc.w 0,0
 dc.b 3,0,0
 dc.b 5
 dc.l koord1
 dc.l 0

koord1:
 dc.w 80,477
 dc.w 500,477
 dc.w 500,490
 dc.w 80,490
 dc.w 80,477

even
AktWin: blk.l 1,0
AktVP: blk.l 1,0

even
Image0:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0111111000000000,%0000000000000000
dc.w %0111111100000000,%0011111000000000
dc.w %0100001100000000,%0011111000000000
dc.w %0100011000000000,%0011110000000000
dc.w %0100001100000000,%0011111000000000
dc.w %0100100110000000,%0011011100000000
dc.w %0011010011000000,%0000001110000000
dc.w %0000001001100000,%0000000111000000
dc.w %0000000100110000,%0000000011100000
dc.w %0000000010100000,%0000000001000000
dc.w %0000000001000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0

even
Image1:
dc.l 0
dc.w %0000001100000000,%0000000000000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000100011000000,%0000011110000000
dc.w %0001000001100000,%0000111111000000
dc.w %0001000111100000,%0000111111000000
dc.w %0000110111000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000010110000000,%0000001100000000
dc.w %0000011110000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0

even
Image2:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000000111111000,%0000000000000000
dc.w %0000001000011000,%0000000111110000
dc.w %0000001000011000,%0000000111110000
dc.w %0000000100011000,%0000000011110000
dc.w %0000001001011000,%0000000111110000
dc.w %0000010011111000,%0000001110110000
dc.w %0000100110110000,%0000011100000000
dc.w %0001001100000000,%0000111000000000
dc.w %0010011000000000,%0001110000000000
dc.w %0001110000000000,%0000100000000000
dc.w %0000100000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0


even
Image3:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000001100000,%0000000000000000
dc.w %0000000010010000,%0000000001100000
dc.w %0111111110001000,%0000000001110000
dc.w %0100000000000100,%0011111111111000
dc.w %0111111111001100,%0011111111111000
dc.w %0111111111011000,%0000000001110000
dc.w %0000000011110000,%0000000001100000
dc.w %0000000001100000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0

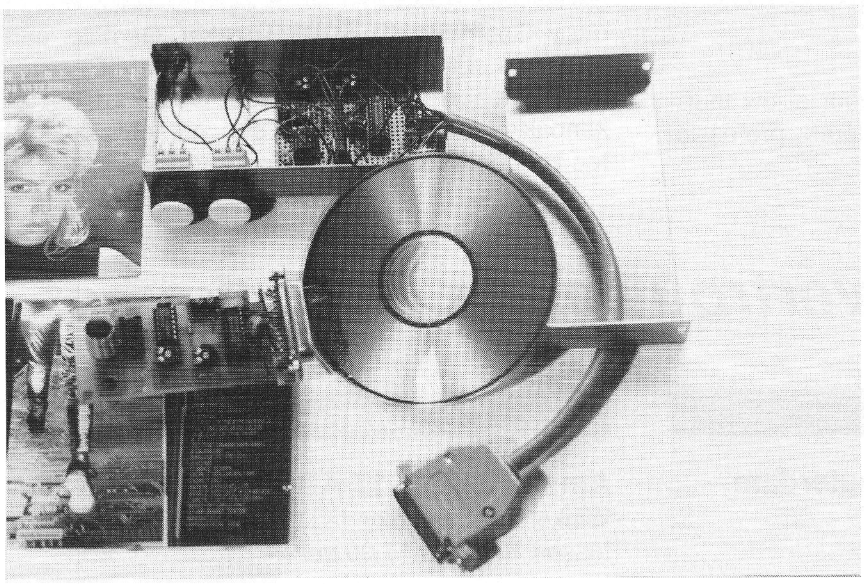
even
Image4:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000100000000000,%0000000000000000
dc.w %0001010000000000,%0000100000000000
dc.w %0011001000000000,%0001110000000000
dc.w %0001100100000000,%0000111000000000
dc.w %0000110010110000,%0000011100000000
dc.w %0000011001001000,%0000001110110000
dc.w %0000001100001000,%0000000111110000
dc.w %0000000110001000,%0000000011110000
dc.w %0000001100001000,%0000000111110000
dc.w %0000001111111000,%0000000111110000
dc.w %0000000111111000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0

even
Image5:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000011110000000,%0000000000000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000111011000000,%0000001100000000
dc.w %0001111000100000,%0000111111000000
dc.w %0001100000100000,%0000111111000000
dc.w %0000110001000000,%0000011110000000
dc.w %0000011010000000,%0000001100000000
dc.w %0000001100000000,%0000000000000000
dc.l 0

even
Image6:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000001000000,%0000000000000000
dc.w %0000000011100000,%0000000001000000
dc.w %0000000110010000,%0000000011100000
dc.w %0000001100100000,%0000000111000000
dc.w %0011011001000000,%0000001110000000
dc.w %0111110010000000,%0011011100000000
dc.w %0110100100000000,%0011111000000000
dc.w %0110001000000000,%0011110000000000
dc.w %0110000100000000,%0011111000000000
dc.w %0110000100000000,%0011111000000000
dc.w %0111111000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0

even
Image7:
dc.l 0
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0001100000000000,%0000000000000000
dc.w %0011110000000000,%0001100000000000
dc.w %0110111111111000,%0011100000000000
dc.w %1100111111111000,%0111111111110000
dc.w %1000000000001000,%0111111111110000
dc.w %0100011111111000,%0011100000000000
dc.w %0010010000000000,%0001100000000000
dc.w %0001100000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.w %0000000000000000,%0000000000000000
dc.l 0
```

Einde listing Spectrum Analyzer



Een zelfgebouwde sampler biedt de mogelijkheid om van elke willekeurige geluidsbron geluiden te digitaliseren.

PRINT OUT Amiga

Form

Form is in december 1988/januari 1989 geschreven door Swiddy de Louw. het programma is vooral bedoeld voor labels voor 3.5" discs, maar door de hoogte en de breedte van het etiket te veranderen kun je het ook voor 5.25" of andere groottes gebruiken.

Deze listing moet je precies zo over tikken als dat hij hier staat behalve als je [RETURN] tegen komt dan moet je op de Returntoets drukken.

Handleiding bij: Form

- Opstarten van Form
Vanuit de CLI typ je "RUN AmigaBASIC FORM" in, Vanuit AmigaBASIC typje "RUN FORM" in en Vanuit de WorkBench kun je dubbelklikken op de Form-icon.Eerst wordt in het programmaatje Form meer geheugen gereserveerd (45Kb)waarna het echte programma wordt geladen. (dit moet FORM.2 heten)

- Openen van een bestand
Om te kunnen werken met Form moet er altijd Eerst een bestand worden geopend.Om dit te doen moet je in de pull-down MENU's "Open bestand" kiezen. Er zal nu worden gevraagd Om een bestand te Openen, je kunt met de muis het gewenstebestand kiezen. Als je een nieuw bestand wilt Openen moet je de naam invoerenDOOR te klikken op het OPEN vlak naast de tekst "File"0! het bestand hier moet altijd een 0!Prefs-bestand zijn.(Form voegt er anders zelf 0!Prefs aan toe)met ACCEPT ga je verder en bevestig je de gekozen naam, met CANCEL ga je weerterug naar het basismenu.
er zal nu Eerst worden gekeken of het gekozen bestand al bestaat, zo ja danwordt er gevraagd of er Als dit nodig is overheen mag worden geschreven.Hiermee kun je voorkomen dat je het 0!Prefs-bestand per ongeluk kwijt raakt.Als alles goed is kom je dan in het Hoofdmenu terecht.

- LAY-OUT veranderen
nu krijgen we wederom een MENU waarbij we kunnen kiezen uit "Preferences" en"Opbouw"0! in de preferences kunnen we instellingen zoals breedte en hoogtevan het etiket veranderen. met gegevensfile wordt het bestand bedoeld waarinalle programma's staan met bijbehorende gegevens zoals soort programma enkleur. met de opbouwfile wordt het bestand bedoeld waarin de tekst van hetlabel staat, dit kun je veranderen via de keuze OPBOUW.
BIJ de keuze opbouw moet je regel voor regel veranderen/toevoegen. het is niet mogelijk Om even terug te gaan naar de eerste regel. per regel is het mogelijk Om de stijl (Condensed, Double WIDTH, etc.), het lettertype,onderlijnen, vet, cursief in te stellen. DOOR [NAAM], [CATEGORIE] of [NUMMER]toe te voegen zal er op die plaats de naam van het programma, het soortprogramma of het NUMMER komen te staan.met "SAVE OPBOUW" wordt de tekst tot en met de huidige regel op disk gezet,met "OK" ga je terug naar het LAY-OUT menu. de volgende keer dat je "OPBOUW"kiest zullen echter de wijzigingen weer worden vergeten. OK is dus zeer tijdelijk. met "CANCEL" worden alle wijzigingen direkt vergeten.

- PRINT-OUT (Uitvoer printer of scherm)
in het MENU wat nu op het beeldscherm komt, kun je kiezen uit printer,beeldscherm, printer verkort en hoofdmenu. Kies je beeldscherm dan wordt het diskette-bestand getoond. Als je printer verkort kiest, dan het diskette-bestand op papier (A4, geen etiketten!) worden afgedrukt.met de keuze printer worden de etiketten gedrukt. Eerst wordt gevraagd of de printer gereed is, waarna er een MENU zal verschijnen waarmee je het begin en eind kunt defenieren. Als je alle labels wilt hebben dan kun je de standaard instelling nemen (1 en 999), anders moet je dit zelf even wijzigen.met de keuze "Begin met printen" zal de printer de etiketten printen.

- pull-down MENU's
in de tweede serie PD-MENU's kunnen we kiezen uit "Informatie", "Kleur-instellingen wijzigen", "Overschrijven (niet) toegestaan" en "(geen) 0!infobestanden creeren". "Informatie" en "Kleurinstellingen wijzigen" behoeven geen nadere uitleg. met de keuze "Overschrijven toegestaan" kun je zorgendat het 0!Prefs-bestand wel of niet overgeschreven mag worden en met ".Infobestanden creeren" kun je ervoor zorgen dat als je bepaalde instellingen(LAY-OUT of opbouw) saved, dat er wel of geen 0!info bestand wordtgecreeerd.

- Hoe zit de gegevensfile in elkaar
dit bestand zul je zelf via een ASCII-Editor moeten ontwerpen. Een voorbeeld:

```
1 ; (altijd een PUNT-KOMMA!!) NUMMER disk
WorkBench 1.3;               naam programma
Systeem;                     CATEGORIE
1 ;                          Kleur

17 ;
BASIC disk;
Programmeer;
2 ;

END

- Hoe zit de opbouwfile in elkaar

80 ; (altijd PUNTKOMMMA) 80=Pica,77=Elite
1 ;1=Normaal,2=Emphas,3=Condensed,4=Double
48; 49 is onderlijnen aan, 48 uit.
72 ; 72 is vet uit, 71 aan.
53 ; 53 is cursief uit, 52 aan.
programma: [naam];    tekst
;
77 ;
4 ;
49 ;
72 ;
53 ;
NUMMER    : [NUMMER];
;
80 ;
1 ;
48 ;
72 ;
53 ;
; Alleen ; geeft aan dat er geen tekst is
END;
```

Listing Form

```
' FORM , geschreven door Swiddy de Louw.
[RETURN]
[RETURN]
' Als u geen VD0: heeft op uw diskette,
vervang dan [RETURN]
' overal in het programma VD0: door RAM:
of DF2:, etc. [RETURN]
[RETURN]
' FORM gebruikt het gehele scherm(PAL),
als uw BASIC niet standaard [RETURN]
' opstart met een PAL-Scherm voeg dan aan
het begin een WINDOW-Statement toe.
[RETURN]
[RETURN]
LIBRARY "intuition.library" [RETURN]
LIBRARY "graphics.library" [RETURN]
LIBRARY "exec.library" [RETURN]
LIBRARY "dos.library" [RETURN]
DECLARE FUNCTION Asksoftstyle& LIBRARY
[RETURN]
DECLARE FUNCTION Setsoftstyle& LIBRARY
[RETURN]
DECLARE FUNCTION Examine& LIBRARY [RETURN]
DECLARE FUNCTION Exnext& LIBRARY [RETURN]
DECLARE FUNCTION Lock& LIBRARY [RETURN]
DECLARE FUNCTION Allocmem& LIBRARY [RETURN]
DECLARE FUNCTION Move& LIBRARY [RETURN]
DECLARE FUNCTION SetWindowTitles& LIBRARY
[RETURN]
DECLARE FUNCTION autorequest& LIBRARY
[RETURN]
DECLARE FUNCTION AllocRemember& LIBRARY
[RETURN]
DIM SHARED
body$(4),dirname$(200),dirblock%(6766),ident
(4,3),double(4,3) [RETURN]
maxidir%=200:taitel$="FORM, geschreven
door Swiddy de Louw." [RETURN]
a=1:ident(1,1)=.135:ident(1,2)=.295:ident(1,
3)=.6 [RETURN]
ident(2,1)=a:ident(2,2)=a:ident(2,3)=a:edl%=
999:stl%=1 [RETURN]
ident(3,1)=0:ident(3,2)=0:ident(3,3)=0
[RETURN]
ident(4,1)=a:ident(4,2)=.5:ident(4,3)=0
[RETURN]
PALETTE 0,ident(1,1),ident(1,2),ident(1,3)
[RETURN]
PALETTE 1,ident(2,1),ident(2,2),ident(2,3)
[RETURN]
PALETTE 2,ident(3,1),ident(3,2),ident(3,3)
[RETURN]
PALETTE 3,ident(4,1),ident(4,2),ident(4,3)
[RETURN]
FOR f=1 TO 4:FOR g=1 TO 3 [RETURN]
double(f,g)=ident(f,g) [RETURN]
NEXT g:NEXT f [RETURN]
titel taitel$,"FORM" [RETURN]
MENU 1,0,1,"Bestand":FOR f=2 TO 9:MENU
f,0,0,"":NEXT [RETURN]
MENU 1,1,1,"Open bestand":MENU
1,2,1,"Verlaat FORM" [RETURN]
CLS:style 6 [RETURN]
LOCATE 10,9:display "FORM is een programma
waarmee labels voor 3.5"+CHR$(34)+"
diskettes" [RETURN]
LOCATE 12,9:display "kunnen worden gedrukt
en waarbij ook rekening wordt gehouden"
[RETURN]
LOCATE 14,9:display "met kleurenprinters.
Het programma is icoongestuurd, dus"
[RETURN]
LOCATE 16,9:display "makkelijk te
gebruiken." [RETURN]
LOCATE 20,13:display "De
programmeur.":style 0 [RETURN]
12: antwoordje=0:ON MENU GOSUB men:MENU ON
[RETURN]
11: [RETURN]
IF antwoordje=0 THEN 11 [RETURN]
IF antwoordje=2 THEN RUN [RETURN]
MENU 1,0,0:ON MENU GOSUB 0 [RETURN]
body$(3)="":yes$=" Ja ":no$=" Nee
":body$(1)="":body$(2)=" Weet u dat zeker
??" [RETURN]
request 250,70,3 [RETURN]
IF res&=0 THEN MENU 1,0,1:GOTO 12 [RETURN]
SYSTEM [RETURN]
men: [RETURN]
IF MENU(1)=2 THEN antwoordje=1:RETURN
[RETURN]
dir 50,35,"Bestand openen.
(Prefs)","DF0:FORM/Bestanden","VD0:"
[RETURN]
IF dirchoice=7 THEN RETURN [RETURN]
IF RIGHT$(program$,6)<>".Prefs" THEN
file$=file$+".Prefs":program$=program$+".Pre
fs" [RETURN]
er&=Lock&(SADD(program$),-2):existing=er&
[RETURN]
CALL Unlock(er&):eroverheen=1 [RETURN]
IF existing>0 THEN [RETURN]
body$(1)=" "+UCASE$(file$):body$(2)="
bestaat al, kan er zo nodig" [RETURN]
body$(3)=" overheen geschreven
worden?":yes$=" Ja ":no$=" Nee " [RETURN]
st$="label":request 290,75,3 [RETURN]
eroverheen=res& [RETURN]
END IF [RETURN]
IF eroverheen=0 THEN over$="niet " ELSE
over$="" [RETURN]
IF existing>0 THEN [RETURN]
OPEN program$ FOR INPUT AS #1 [RETURN]
LINE INPUT#1,a$:nlqofdraft%=VAL(a$)
[RETURN]
LINE INPUT#1,a$:space%=VAL(a$) [RETURN]
LINE INPUT#1,a$:breedte%=VAL(a$) [RETURN]
LINE INPUT#1,a$:hoogte%=VAL(a$) [RETURN]
LINE INPUT#1,a$:lmarge%=VAL(a$) [RETURN]
LINE
INPUT#1,a$:opbouw$=LEFT$(a$,INSTR(a$,";")-1)
[RETURN]
LINE
INPUT#1,a$:bestand$=LEFT$(a$,INSTR(a$,";")-1
) [RETURN]
CLOSE #1 [RETURN]
END IF [RETURN]
bron$=opbouw$:puntinfo$="Geen ":MENU
1,0,0:MENU 2,0,1,"Tools":MENU
2,1,1,"Informatie                    "
[RETURN]
MENU 2,2,1,"Kleurinstellingen
":ON MENU GOSUB afhandel2 [RETURN]
MENU 2,3,1,"Overschrijven
"+over$+"toegestaan" [RETURN]
MENU 2,4,1,puntinfo$+".Info bestanden
creeren " [RETURN]
MENU 2,5,1,"Positie printkop
" [RETURN]
```

```
create2 7:create6 3:create10 4:create11
1,4:create8 8:create 11 [RETURN]
masterloep: CLS:titel taitel$,"FORM"
[RETURN]
MENU ON:define2 1,150,76,260,14,"Verander
de lay-out.",2,1,1 [RETURN]
define2 2,150,92,260,14,"Uitvoer naar
printer of scherm.",2,1,1 [RETURN]
define2 3,150,108,260,14,"Naar het
basismenu.",3,1,2 [RETURN]
define6 1,50,6,200,"RGB 1",0,1,0,0,2,1,1
[RETURN]
define6 2,50,18,200,"RGB 2",0,1,0,0,2,1,1
[RETURN]
define6 3,50,30,200,"RGB 3",0,1,0,0,2,1,1
[RETURN]
show2 3:style 2:texting 220,66,"FORM
Hoofdmenu." [RETURN]
style 6:texting 30,20,"Lay-out bestand:
":style 2:display
program$:choicemade=0:style 0 [RETURN]
WHILE choicemade=0 [RETURN]
WHILE MOUSE(0)>=0:WEND [RETURN]
control2 3 [RETURN]
WEND [RETURN]
IF choicemade=3 THEN  [RETURN]
body$(1)=" WAARSCHUWING: Alle huidige"
[RETURN]
body$(2)=" gegevens zullen worden" [RETURN]
body$(3)=" gewist !!":no$=" Cancel
":yes$="  Ok  " [RETURN]
request 266,75,3 [RETURN]
IF res&=1 THEN MENU 1,0,1:MENU
2,0,0:antwoordje=2:RETURN [RETURN]
END IF [RETURN]
ON choicemade GOSUB verander,uitvoer
[RETURN]
GOTO masterloep [RETURN]
RETURN [RETURN]
afhandel2: [RETURN]
IF MENU(1)=1 AND MENU(0)=2 THEN [RETURN]
no$=" Continue ":yes$=no$:body$(1)=" FORM
is in december 1988" [RETURN]
body$(2)=" geschreven door Swiddy de
Louw." [RETURN]
st$="FORM":request 320,70,2 [RETURN]
ELSEIF MENU(1)=2 THEN [RETURN]
WINDOW 2,"Kleurinstellingen
wijzigen.",(30,15)-(365,70),18 [RETURN]
ok=0:klid=1:varia6(1)=ident(1,1):varia6(2)=i
dent(1,2):varia6(3)=ident(1,3):show6 3
[RETURN]
LINE (50,43)-(112,49),0,bf [RETURN]
LINE (113,43)-(177,49),1,bf [RETURN]
LINE (178,43)-(242,49),2,bf [RETURN]
LINE (243,43)-(310,49),3,bf [RETURN]
LINE (270,6)-(310,21),2,bf [RETURN]
LINE (270,6)-(310,21),1,b [RETURN]
LINE (270,26)-(310,40),2,bf [RETURN]
LINE (270,26)-(310,40),1,b [RETURN]
LINE (50,43)-(310,49),1,b [RETURN]
LINE (277,29)-(301,37),1,b [RETURN]
LINE (277,29)-(301,37),1 [RETURN]
LINE (277,37)-(301,29),1 [RETURN]
COLOR 1,2:texting 280,17,"OK!":COLOR 1,0
[RETURN]
WHILE ok=0 [RETURN]
WHILE MOUSE(0)>=0:WEND [RETURN]
control6 3 [RETURN]
PALETTE
klid-1,varia6(1),varia6(2),varia6(3)
[RETURN]
x=MOUSE(1):y=MOUSE(2) [RETURN]
IF x>269 AND x<311 AND y>5 AND y<22 THEN
ok=1 [RETURN]
IF x>269 AND x<311 AND y>25 AND y<41 THEN
ok=2 [RETURN]
IF x>49 AND x<311 AND y>42 AND y<50 THEN
[RETURN]
f=klid:IF x<113 THEN klid=1 [RETURN]
IF x>112 AND x<178 THEN klid=2 [RETURN]
IF x>177 AND x<243 THEN klid=3 [RETURN]
IF x>242 THEN klid=4 [RETURN]
IF f<>klid THEN [RETURN]
ident(f,1)=varia6(1) [RETURN]
ident(f,2)=varia6(2) [RETURN]
ident(f,3)=varia6(3) [RETURN]
varia6(1)=ident(klid,1) [RETURN]
varia6(2)=ident(klid,2) [RETURN]
varia6(3)=ident(klid,3) [RETURN]
show6 3:PALETTE
klid-1,varia6(1),varia6(2),varia6(3)
[RETURN]
END IF [RETURN]
END IF [RETURN]
FOR g=1 TO 400:NEXT [RETURN]
WEND [RETURN]
IF ok=2 THEN [RETURN]
FOR f=1 TO 4:FOR g=1 TO 3 [RETURN]
ident(f,g)=double(f,g) [RETURN]
NEXT g:NEXT f [RETURN]
PALETTE 0,ident(1,1),ident(1,2),ident(1,3)
[RETURN]
PALETTE 1,ident(2,1),ident(2,2),ident(2,3)
[RETURN]
PALETTE 2,ident(3,1),ident(3,2),ident(3,3)
[RETURN]
PALETTE 3,ident(4,1),ident(4,2),ident(4,3)
[RETURN]
ELSE [RETURN]
ident(klid,1)=varia6(1) [RETURN]
ident(klid,2)=varia6(2) [RETURN]
ident(klid,3)=varia6(3) [RETURN]
FOR f=1 TO 4:FOR g=1 TO 3 [RETURN]
double(f,g)=ident(f,g) [RETURN]
NEXT g:NEXT f [RETURN]
END IF [RETURN]
WINDOW CLOSE 2 [RETURN]
ELSEIF MENU(1)=3 THEN [RETURN]
eroverheen=(1 AND eroverheen=0) [RETURN]
IF eroverheen=1 THEN over$="" ELSE
over$="niet " [RETURN]
MENU 2,3,1,"Overschrijven
"+over$+"toegestaan" [RETURN]
ELSEIF MENU(1)=4 THEN [RETURN]
IF puntinfo$="Geen " THEN puntinfo$=""
ELSE puntinfo$="Geen " [RETURN]
MENU 2,4,1,puntinfo$+".Info bestanden
creeren " [RETURN]
ELSEIF MENU(1)=5 THEN [RETURN]
no$=" Continue ":yes$=no$ [RETURN]
body$(1)=" De printkop moet zo staan, dat
het lint" [RETURN]
body$(2)=" precies tegenover de bovenkant
van het" [RETURN]
body$(3)=" etiket staat." [RETURN]
request 370,75,3  [RETURN]
END IF [RETURN]
RETURN [RETURN]
```

```
verander:         ' verander de lay-out
[RETURN]
CLS:taitel$="FORM, geschreven door Swiddy
de Louw.              LAY-OUT" [RETURN]
titel taitel$,"FORM" [RETURN]
define2 1,150,76,125,32,"
PREFERENCES",2,1,1 [RETURN]
define2 2,280,76,125,32,"      OPBOUW",2,1,1
[RETURN]
define2 3,150,112,255,15,"     TERUG NAAR
HET HOOFDMENU",3,1,2 [RETURN]
LINE (95,22)-(475,165),1,b [RETURN]
show2 3:choicemade=0 [RETURN]
style 6:texting 102,35,"In dit gedeelte
kunnen de instellingen zoals" [RETURN]
texting 102,45,"breedte, hoogte
(Preferences) en de tekst van" [RETURN]
texting 102,55,"van het label (Opbouw)
worden gedefenieerd." [RETURN]
texting 102,160,"Lay-out bestand: ":style
2:display file$ [RETURN]
style 0 [RETURN]
WHILE choicemade=0 [RETURN]
WHILE MOUSE(0)>=0:WEND [RETURN]
control2 3 [RETURN]
WEND [RETURN]
IF choicemade=3 THEN RETURN [RETURN]
ON choicemade GOSUB prefs,opbouw [RETURN]
FOR f=1 TO 11:controllen(f,0)=-1:NEXT
[RETURN]
GOTO verander [RETURN]
RETURN [RETURN]
prefs: [RETURN]
CLS:define10 1,20,10,355,12," Ruimte
tussen 2 labels in
milimeters",space%,0,99,3,3,1,2 [RETURN]
define10 2,20,25,355,12,"Breedte van een
label in
milimeters",breedte%,0,999,4,2,1,1 [RETURN]
define10 3,20,40,355,12,"Hoogte van een
label in milimeters",hoogte%,0,999,4,2,1,1
[RETURN]
define10 4,20,55,355,12," Marge van een
label in karakters",lmarge%,0,99,3,1,2,2
[RETURN]
define2 1,20,70,355,12," Bestandsnaam van
de opbouwfile:",2,1,0 [RETURN]
define2 2,20,85,355,12,"",2,1,0 [RETURN]
define2 3,20,100,355,12," Bestandsnaam van
de gegevensfile:",2,1,3 [RETURN]
define2 4,20,115,355,12,"",2,1,0 [RETURN]
define11 1,20,130,58,2,nlqofdraft%+1,2,1,3
[RETURN]
text11$(1,1)="Draft":text11$(1,2)=" NLQ"
[RETURN]
define2 5,90,130,285,24,"          SAVE
INSTELLINGEN",1,2,2 [RETURN]
define2 6,20,157,170,40,"
OK",3,2,2 [RETURN]
define2 7,198,157,177,40,"
CANCEL",3,2,2 [RETURN]
show10 4:show2 7:show11 1 [RETURN]
LINE (400,10)-(600,197),1,b:style 6
[RETURN]
texting 408,26,"Een karakter op papier"
[RETURN]
texting 408,37,"heeft een breedte van 3"
[RETURN]
texting 408,48,"en een hoogte van 4 mm."
[RETURN]
texting 408,59,"Dus op een label van 72"
[RETURN]
texting 408,70,"bij 72 mm, passen acht-"
[RETURN]
texting 408,81,"tien regels tekst en"
[RETURN]
texting 408,92,"per regel 24 karakters"
[RETURN]
texting 408,103,"in Pica (10 cpi)" [RETURN]
texting 408,114,"Met ~opbouwfile~ wordt"
[RETURN]
texting 408,125,"bedoeld het bestand"
[RETURN]
texting 408,136,"waarin de tekst van"
[RETURN]
texting 408,147,"labels staat.In de"
[RETURN]
texting 408,158,"~gegevensfile~ staan"
[RETURN]
texting 408,169,"bijvoorbeeld alle"
[RETURN]
texting 408,180,"programmanamen," [RETURN]
texting 408,191,"nummers, enz." [RETURN]
texting 20,220,"Lay-out bestand: ":display
program$:style 0 [RETURN]
define 2,20,70,355,127,7 [RETURN]
define 10,20,10,355,67,4 [RETURN]
define 11,20,130,58,24,1 [RETURN]
zolang=0:bouwop$=opbouw$:gegeven$=bestand$
[RETURN]
COLOR 1,2:texting
36,94,LEFT$(bouwop$,40):texting
36,124,LEFT$(gegeven$,40):COLOR 1,0
[RETURN]
WHILE zolang=0 [RETURN]
master 11 [RETURN]
IF choicemade=1 OR choicemade=2 THEN
[RETURN]
bet1$=program$:bet2$=file$:bet3$=drawer$
[RETURN]
drawer$="DF0:Examples":file$="":dir
50,35,"Opbouwfile.","DF0:Examples","VD0:"
[RETURN]
IF dirchoice=6 THEN bouwop$=program$
[RETURN]
program$=bet1$:file$=bet2$:drawer$=bet3$
[RETURN]
LINE (21,86)-(374,96),2,bf [RETURN]
COLOR 1,2:texting
36,94,LEFT$(bouwop$,40):COLOR 1,0 [RETURN]
ELSEIF choicemade=3 OR choicemade=4 THEN
[RETURN]
bet1$=program$:bet2$=file$:bet3$=drawer$
[RETURN]
drawer$="DF0:Examples":file$="":dir
50,35,"Gegevensfile.","DF0:Examples","VD0:"
[RETURN]
IF dirchoice=6 THEN gegeven$=program$
[RETURN]
program$=bet1$:file$=bet2$:drawer$=bet3$
[RETURN]
LINE (21,116)-(374,126),2,bf [RETURN]
COLOR 1,2:texting
36,124,LEFT$(gegeven$,40):COLOR 1,0
[RETURN]
ELSEIF choicemade=5 THEN [RETURN]
IF eroverheen=1 THEN [RETURN]
OPEN program$ FOR OUTPUT AS #2 [RETURN]
PRINT#2,STR$(position11(1,5)-1)+";"
[RETURN]
PRINT#2,STR$(value10(1))+";" [RETURN]
```

```
PRINT#2,STR$(value10(2))+";" [RETURN]
PRINT#2,STR$(value10(3))+";" [RETURN]
PRINT#2,STR$(value10(4))+";" [RETURN]
PRINT#2,bouwop$+";" [RETURN]
PRINT#2,gegeven$+";" [RETURN]
CLOSE#2 [RETURN]
IF puntinfo$="Geen " THEN KILL
program$+".info" [RETURN]
ELSE [RETURN]
body$(1)=" Overschrijven van" [RETURN]
body$(2)=" "+file$ [RETURN]
body$(3)=" niet toegestaan":no$=" Continue
":yes$=no$ [RETURN]
st$="FORM":request 286,75,3 [RETURN]
END IF [RETURN]
ELSEIF choicemade=7 THEN [RETURN]
zolang=1 [RETURN]
ELSE [RETURN]
zolang=1 [RETURN]
nlqofdraft%=position11(1,5)-1 [RETURN]
space%=value10(1):breedte%=value10(2)
[RETURN]
hoogte%=value10(3):lmarge%=value10(4)
[RETURN]
opbouw$=bouwop$:bestand$=gegeven$ [RETURN]
END IF [RETURN]
WEND [RETURN]
RETURN [RETURN]
opbouw: [RETURN]
IF hoogte%=0 THEN RETURN    ' grootte,
breedte van label moet bekend zijn [RETURN]
CLS:define11 1,20,10,112,4,1,2,1,3 [RETURN]
text11$(1,1)="Normaal":text11$(1,2)="Emphasi
zed" [RETURN]
text11$(1,3)="Condensed":text11$(1,4)="Doubl
e Width" [RETURN]
define8 1,249,10,56,12,"Pica",1,2,1,3
[RETURN]
define8 2,305,10,56,12,"Elite",0,2,1,3
[RETURN]
react8(1,2)=3:react8(2,1)=3 [RETURN]
define8 3,249,22,56,12," AAN",0,2,1,3
[RETURN]
define8 4,305,22,56,12," UIT",1,2,1,3
[RETURN]
react8(3,4)=3:react8(4,3)=3 [RETURN]
define8 5,249,34,56,12," AAN",0,2,1,3
[RETURN]
define8 6,305,34,56,12," UIT",1,2,1,3
[RETURN]
react8(5,6)=3:react8(6,5)=3 [RETURN]
define8 7,249,46,56,12," AAN",0,2,1,3
[RETURN]
define8 8,305,46,56,12," UIT",1,2,1,3
[RETURN]
react8(7,8)=3:react8(8,7)=3        [RETURN]
define2 1,442,10,120,48," SAVE
OPBOUW",3,2,2 [RETURN]
define2 2,442,68,120,48,"        OK",3,2,2
[RETURN]
define2 3,442,126,120,48,"
CANCEL",3,2,2 [RETURN]
define2 4,20,68,341,48,"",2,1,1 [RETURN]
define2 5,20,126,341,48,"
VOLGENDE REGEL",2,1,1 [RETURN]
show11 1:show8 8:show2 5 [RETURN]
texting 153,20,"Lettertype" [RETURN]
texting 153,32,"Onderlijnen" [RETURN]
texting 153,44,"Vet" [RETURN]
texting 153,56,"Cursief"  [RETURN]
LINE (20,184)-(562,232),2,bf:LINE
(20,184)-(562,232),1,b:style 6:COLOR 3,2
[RETURN]
texting 28,196,"Opbouw-bestand: "+opbouw$
[RETURN]
texting 28,212,"Opmerking: Het is veel
gemakkelijker om het ~opbouw~-bestand via"
[RETURN]
texting 28,223,"een ASCII-Editor te
veranderen. (bv. TxED of ED)":COLOR 1,0
[RETURN]
style 0:LINE (36,85)-(345,97),0,bf [RETURN]
LINE (35,84)-(346,98),1,b:COLOR
1,2:texting 44,81,"Voer hier de tekst in:"
[RETURN]
texting 44,110,"[NAAM]         [CATEGORIE]
[NUMMER]":COLOR 1,0 [RETURN]
define 11,20,10,112,48,1 [RETURN]
define 8,249,10,361,48,8 [RETURN]
define 2,20,10,562,177,5 [RETURN]
cpr%=INT(breedte%/3):lth%=INT(hoogte%/4)
[RETURN]
er&=Lock&(SADD(opbouw$),-2):exi=er&:CALL
Unlock(er&) [RETURN]
IF exi>0 THEN OPEN opbouw$ FOR INPUT AS #1
[RETURN]
OPEN "vd0:FORM.OPB" FOR OUTPUT AS #2
[RETURN]
regel=1:zolang=0:lastline=0:endje=0
[RETURN]
lettyp%=80:stijl%=1:onderlijn%=48:vet%=72:cu
rsief%=53 [RETURN]
WHILE zolang=0 [RETURN]
IF exi>0 THEN [RETURN]
LINE INPUT#1,a$:lettyp%=VAL(a$) [RETURN]
LINE INPUT#1,a$:stijl%=VAL(a$) [RETURN]
LINE INPUT#1,a$:onderlijn%=VAL(a$) [RETURN]
LINE INPUT#1,a$:vet%=VAL(a$) [RETURN]
LINE INPUT#1,a$:cursief%=VAL(a$) [RETURN]
LINE INPUT#1,a$:tkst$=LEFT$(a$,LEN(a$)-1)
[RETURN]
LINE INPUT#1,a$:IF UCASE$(a$)="END;" THEN
lastline=1 [RETURN]
ELSE [RETURN]
tkst$="" [RETURN]
END IF [RETURN]
position11(1,5)=stijl% [RETURN]
IF lettyp%=80 THEN stand8(1)=1:stand8(2)=0
ELSE stand8(2)=1:stand8(1)=1 [RETURN]
IF onderlijn%=49 THEN
stand8(3)=1:stand8(4)=0 ELSE
stand8(4)=1:stand8(3)=0 [RETURN]
IF vet%=71 THEN stand8(5)=1:stand8(6)=0
ELSE stand8(6)=1:stand8(5)=0 [RETURN]
IF cursief%=52 THEN
stand8(7)=1:stand8(8)=0 ELSE
stand8(8)=1:stand8(7)=0 [RETURN]
LINE (36,85)-(345,97),0,bf:texting
44,94,LEFT$(tkst$,36) [RETURN]
show11 1:show8 8
[RETURN]
q2: [RETURN]
titel taitel$,"FORM" [RETURN]
master 11 [RETURN]
IF choicemade=1 THEN [RETURN]
st$="FORM":body$(1)=" Mag er over het
opbouw-bestand" [RETURN]
body$(2)=" heen worden geschreven
?":yes$=" Ja ":no$=" Nee " [RETURN]
request 300,65,2 [RETURN]
IF res&=1 THEN [RETURN]
```

```
endje=1:plaats [RETURN]
dorest [RETURN]
OPEN "VD0:FORM.OPB" FOR INPUT AS #1
[RETURN]
OPEN opbouw$ FOR OUTPUT AS #2 [RETURN]
dorest [RETURN]
zolang=1:bron$=opbouw$ [RETURN]
IF puntinfo$="Geen " THEN KILL
opbouw$+".info" [RETURN]
END IF [RETURN]
ELSEIF choicemade=2 THEN [RETURN]
body$(1)=" Bij de volgende aanroep van
OPBOUW" [RETURN]
body$(2)=" zal het ongewijzigde bestand
weer" [RETURN]
body$(3)=" worden gebruikt. Is dit goed ?"
[RETURN]
yes$=" Ja ":no$=" Nee ":request 347,76,3
[RETURN]
IF res&=0 THEN [RETURN]
GOTO q2 [RETURN]
ELSE [RETURN]
endje=1:plaats [RETURN]
dorest [RETURN]
zolang=1:bron$="VD0:FORM.OPB" [RETURN]
END IF [RETURN]
ELSEIF choicemade=3 THEN [RETURN]
CLOSE#2:KILL "vd0:FORM.OPB" [RETURN]
zolang=1:bron$=opbouw$ [RETURN]
ELSEIF choicemade=4 THEN [RETURN]
ed tkst$,36,cpr%+50,44,94 [RETURN]
tkst$=inv$:LINE
(36,85)-(345,97),0,bf:texting
44,94,LEFT$(tkst$,36) [RETURN]
GOTO q2 [RETURN]
ELSEIF choicemade=5 THEN [RETURN]
IF regel=lth% THEN endje=1 [RETURN]
IF regel>lth% THEN [RETURN]
BEEP:GOTO q2 [RETURN]
ELSEIF lastline=0 THEN [RETURN]
regel=regel+1:plaats [RETURN]
ELSEIF lastline=1 AND regel+1<=lth% THEN
[RETURN]
exi=0:regel=regel+1:plaats [RETURN]
END IF [RETURN]
IF endje=1 THEN BEEP [RETURN]
END IF [RETURN]
WEND [RETURN]
CLOSE#2:CLOSE#1 [RETURN]
RETURN [RETURN]
SUB titel (wt$,st$) STATIC [RETURN]
wt$=wt$+CHR$(0):st$=st$+CHR$(0) [RETURN]
CALL
SetWindowTitles(WINDOW(7),SADD(wt$),SADD(st$
)) [RETURN]
END SUB [RETURN]
SUB create2 (num%) STATIC [RETURN]
DIM SHARED
position2(num%,4),text2$(num%),colour2(num%,
3) [RETURN]
END SUB [RETURN]
SUB define2
(num%,xpos%,ypos%,xlength%,ylength%,text$,co
l1%,col2%,col3%) STATIC [RETURN]
position2(num%,1)=xpos% [RETURN]
position2(num%,2)=ypos% [RETURN]
position2(num%,3)=xlength% [RETURN]
position2(num%,4)=ylength% [RETURN]
text2$(num%)=text$ [RETURN]
colour2(num%,1)=col1% [RETURN]
colour2(num%,2)=col2% [RETURN]
colour2(num%,3)=col3% [RETURN]
END SUB [RETURN]
SUB show2 (num%) STATIC [RETURN]
FOR f=1 TO num% [RETURN]
COLOR colour2(f,3),colour2(f,1) [RETURN]
LINE
(position2(f,1),position2(f,2))-(position2(f
,1)+position2(f,3),position2(f,2)+position2(
f,4)),colour2(f,1),bf [RETURN]
q=(position2(f,4)-8)/2:b%=position2(f,1)+8:a
%=position2(f,2)+q+8:texting
b%,a%,text2$(f) [RETURN]
LINE
(position2(f,1),position2(f,2))-(position2(f
,1)+position2(f,3),position2(f,2)+position2(
f,4)),colour2(f,2),b [RETURN]
NEXT [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB control2 (num%) STATIC [RETURN]
SHARED choicemade [RETURN]
x=MOUSE(1):y=MOUSE(2) [RETURN]
choicemade=0 [RETURN]
FOR f=1 TO num% [RETURN]
xway=0:yway=0 [RETURN]
IF x>position2(f,1)-1 AND
x<position2(f,1)+position2(f,3)+1 THEN
xway=1 [RETURN]
IF y>position2(f,2)-2 AND
y<position2(f,2)+position2(f,4) THEN
yway=1 [RETURN]
IF xway=1 AND yway=1 THEN
choicemade=f:f=num% [RETURN]
NEXT f [RETURN]
END SUB [RETURN]
SUB create10 (n%) STATIC [RETURN]
DIM SHARED
position10(n%,4),limit10(n%,3),text10$(n%),c
olour10(n%,3),value10(n%) [RETURN]
END SUB [RETURN]
SUB define10
(n%,x1%,y1%,x2%,y2%,tekst$,v%,l1%,l2%,l3%,c1
%,c2%,c3%) STATIC [RETURN]
position10(n%,1)=x1%:position10(n%,2)=y1%
[RETURN]
position10(n%,3)=x2%:position10(n%,4)=y2%
[RETURN]
limit10(n%,1)=l1%:limit10(n%,2)=l2%
[RETURN]
limit10(n%,3)=l3%:value10(n%)=v% [RETURN]
text10$(n%)=tekst$:colour10(n%,1)=c1%
[RETURN]
colour10(n%,2)=c2%:colour10(n%,3)=c3%
[RETURN]
END SUB [RETURN]
SUB show10 (n%) STATIC [RETURN]
FOR f=1 TO n% [RETURN]
COLOR colour10(f,3),colour10(f,1) [RETURN]
LINE
(position10(f,1),position10(f,2))-(position1
0(f,1)+position10(f,3),position10(f,2)+posit
ion10(f,4)),colour10(f,1),bf [RETURN]
LINE
(position10(f,1),position10(f,2))-(position1
0(f,1)+position10(f,3),position10(f,2)+posit
ion10(f,4)),colour10(f,2),b [RETURN]
a%=24+position10(f,1)+(limit10(f,3)*8):b%=po
sition10(f,2)+3+(position10(f,4)/2)
[RETURN]
```

```
texting
a%,b%,text10$(f):a%=8+position10(f,1):textin
g a%,b%,STR$(value10(f)) [RETURN]
NEXT f [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB control10 (n%) STATIC [RETURN]
SHARED choicemade,inv,inv$ [RETURN]
x%=MOUSE(1):y%=MOUSE(2):choicemade=0
[RETURN]
FOR f=1 TO n% [RETURN]
xway=0:yway=0 [RETURN]
IF x%>position10(f,1)-1 AND
x%<position10(f,1)+position10(f,3)+1 THEN
xway=1 [RETURN]
IF y%>position10(f,2)-1 AND
y%<position10(f,2)+position10(f,4)+1 THEN
yway=1 [RETURN]
IF xway=1 AND yway=1 THEN
choicemade=f:f=n% [RETURN]
NEXT f [RETURN]
f=choicemade:IF f=0 THEN EXIT SUB [RETURN]
a%=16+position10(f,1):b%=position10(f,2)+3+(
position10(f,4)/2):c%=limit10(f,3) [RETURN]
COLOR
colour10(f,3),colour10(f,1):inv=-32768&:text
ing a%-8,b%," " [RETURN]
WHILE NOT(inv>limit10(f,1)-1 AND
inv<limit10(f,2)+1) [RETURN]
ed STR$(value10(f)),c%,c%+1,a%,b% [RETURN]
WEND [RETURN]
value10(f)=inv:verwerk
a%-8,b%,STR$(inv),c%+1:COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB create11 (n%,m%) STATIC [RETURN]
DIM SHARED
position11(n%,5),text11$(n%,m%),colour11(n%,
3) [RETURN]
END SUB [RETURN]
SUB define11
(n%,x1%,y%,x2%,num%,v%,c1%,c2%,c3%) STATIC
[RETURN]
position11(n%,1)=x1%:position11(n%,2)=y%:pos
ition11(n%,3)=x2% [RETURN]
position11(n%,4)=num%:position11(n%,5)=v%
[RETURN]
colour11(n%,1)=c1%:colour11(n%,2)=c2%:colour
11(n%,3)=c3% [RETURN]
END SUB [RETURN]
SUB show11 (n%) STATIC [RETURN]
FOR f=1 TO n% [RETURN]
COLOR
colour11(f,3),colour11(f,1):x%=position11(f,
1):y%=position11(f,2):num%=position11(f,4)
[RETURN]
LINE
(x%,y%)-(x%+position11(f,3),y%+(num%*12)),co
lour11(f,1),bf [RETURN]
LINE
(x%,y%)-(x%+position11(f,3),y%+(num%*12)),co
lour11(f,2),b [RETURN]
FOR g%=1 TO num% [RETURN]
IF g%=position11(f,5) THEN [RETURN]
LINE
(1+x%,1+y%+((g%-1)*12))-(x%+position11(f,3)-
1,y%+(g%*12)),colour11(f,3),bf [RETURN]
COLOR colour11(f,1),colour11(f,3) [RETURN]
END IF [RETURN]
LINE
(x%,y%+(g%*12))-(x%+position11(f,3),y%+(g%*1
2)),colour11(f,2) [RETURN]
texting x%+8,y%-3+(12*g%),text11$(f,g%)
[RETURN]
IF g%=position11(f,5) THEN COLOR
colour11(f,3),colour11(f,1) [RETURN]
NEXT g% [RETURN]
NEXT f [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB control11 (n%) STATIC [RETURN]
SHARED choicemade [RETURN]
x%=MOUSE(1):y%=MOUSE(2):choicemade=0
[RETURN]
FOR f=1 TO n% [RETURN]
xway=0:yway=0 [RETURN]
IF x%>position11(f,1)-1 AND
x%<position11(f,1)+position11(f,3)+1 THEN
xway=1 [RETURN]
IF y%>position11(f,2)-1 AND
y%<position11(f,2)+(position11(f,4)*12)+1
THEN yway=1 [RETURN]
IF xway=1 AND yway=1 THEN
choicemade=f:f=n% [RETURN]
NEXT f [RETURN]
IF choicemade=0 THEN EXIT SUB [RETURN]
c=choicemade:p3%=position11(c,1):p4%=positio
n11(c,2) [RETURN]
FOR f=1 TO position11(c,4) [RETURN]
yway=0:IF y%>p4%+((f-1)*12)-1 AND
y%<p4%+(f*12)+1 THEN yway=1 [RETURN]
IF yway=1 THEN choice=f:f=position11(c,4)
[RETURN]
NEXT [RETURN]
IF position11(c,5)=choice THEN EXIT SUB
[RETURN]
f%=position11(c,5):p%=position11(c,1):p2%=po
sition11(c,2) [RETURN]
LINE
(1+p%,1+p2%+((f%-1)*12))-(p%+position11(c,3)
-1,p2%+(f%*12)-1),colour11(c,1),bf [RETURN]
COLOR colour11(c,3),colour11(c,1):texting
p%+8,p2%-3+(12*f%),text11$(c,position11(c,5)
) [RETURN]
f%=choice:p%=p3%:p2%=p4% [RETURN]
LINE
(1+p%,1+p2%+((f%-1)*12))-(p%+position11(c,3)
-1,p2%+(f%*12)-1),colour11(c,3),bf [RETURN]
COLOR colour11(c,1),colour11(c,3):texting
p%+8,p2%-3+(12*f%),text11$(c,choice)
[RETURN]
position11(c,5)=choice [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB create8 (n%) STATIC [RETURN]
DIM SHARED
position8(n%,4),text8$(n%),colour8(n%,3),sta
nd8(n%),react8(n%,n%) [RETURN]
END SUB [RETURN]
SUB define8
(n%,x%,y%,x1%,y1%,tekst$,st%,c1%,c2%,c3%)
STATIC [RETURN]
position8(n%,1)=x%:position8(n%,2)=y%
[RETURN]
position8(n%,3)=x1%:position8(n%,4)=y1%
[RETURN]
text8$(n%)=tekst$:colour8(n%,1)=c1%
[RETURN]
colour8(n%,2)=c2%:colour8(n%,3)=c3%:stand8(n
%)=st% [RETURN]
END SUB [RETURN]
SUB show8 (n%) STATIC [RETURN]
FOR f=1 TO n% [RETURN]
```

```
COLOR colour8(f,3),colour8(f,1) [RETURN]
LINE
(position8(f,1),position8(f,2))-(position8(f
,1)+position8(f,3),position8(f,2)+position8(
f,4)),colour8(f,1),bf [RETURN]
LINE
(position8(f,1),position8(f,2))-(position8(f
,1)+position8(f,3),position8(f,2)+position8(
f,4)),colour8(f,2),b [RETURN]
txy%=position8(f,2)+((position8(f,4)-8)/2)+8
:txx%=position8(f,1)+8:texting
txx%,txy%,text8$(f) [RETURN]
IF stand8(f)=0 THEN LINE
(position8(f,1)+1,position8(f,2)+position8(f
,4)-1)-(position8(f,1)+position8(f,3)-1,posi
tion8(f,2)+1),colour8(f,3) [RETURN]
NEXT f [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB control8 (num%) STATIC [RETURN]
SHARED choicemade,priority [RETURN]
x=MOUSE(1):y=MOUSE(2) [RETURN]
choicemade=0 [RETURN]
FOR f=1 TO num% [RETURN]
xway=0:yway=0 [RETURN]
IF x>position8(f,1)-2 AND
x<position8(f,1)+position8(f,3) THEN
xway=1 [RETURN]
IF y>position8(f,2)-2 AND
y<position8(f,2)+position8(f,4) THEN
yway=1 [RETURN]
IF xway=1 AND yway=1 THEN
choicemade=f:f=num% [RETURN]
NEXT f [RETURN]
IF choicemade=0 THEN EXIT SUB [RETURN]
g=choicemade:changes=1 [RETURN]
IF priority>0 AND g<>priority AND
stand8(priority)=0 THEN EXIT SUB [RETURN]
FOR f=1 TO num%
[RETURN]
IF react8(g,f)=0 THEN NEKST [RETURN]
IF react8(g,f)=2 AND stand8(f)=0 THEN
f=num%:changes=-1:GOTO NEKST [RETURN]
IF react8(g,f)=-2 AND stand8(f)=1 THEN
f=num%:changes=-1 [RETURN]
NEKST: NEXT f [RETURN]
IF changes=-1 THEN EXIT SUB [RETURN]
DIM shadow(num%):FOR f=1 TO
num%:shadow(f)=stand8(f):NEXT f [RETURN]
IF stand8(g)=1 THEN stand8(g)=0 ELSE
stand8(g)=1 [RETURN]
FOR f=1 TO num% [RETURN]
IF  react8(g,f)=1 THEN [RETURN]
stand8(f)=1 [RETURN]
ELSEIF react8(g,f)=-1 THEN  [RETURN]
stand8(f)=0 [RETURN]
ELSEIF react8(g,f)=3 THEN [RETURN]
IF stand8(g)=1 THEN stand8(f)=0 ELSE
stand8(f)=1 [RETURN]
END IF [RETURN]
NEXT f [RETURN]
FOR f=1 TO num% [RETURN]
IF shadow(f)=1 AND stand8(f)=0 THEN
[RETURN]
LINE
(position8(f,1)+1,position8(f,2)+position8(f
,4)-1)-(position8(f,1)+position8(f,3)-1,posi
tion8(f,2)+1),colour8(f,3)  [RETURN]
ELSEIF shadow(f)=0 AND stand8(f)=1 THEN
[RETURN]
COLOR colour8(f,3),colour8(f,1) [RETURN]
LINE
(position8(f,1),position8(f,2))-(position8(f
,1)+position8(f,3),position8(f,2)+position8(
f,4)),colour8(f,1),bf [RETURN]
LINE
(position8(f,1),position8(f,2))-(position8(f
,1)+position8(f,3),position8(f,2)+position8(
f,4)),colour8(f,2),b  [RETURN]
txy%=position8(f,2)+((position8(f,4)-8)/2)+8
:txx%=position8(f,1)+8:texting
txx%,txy%,text8$(f) [RETURN]
END IF [RETURN]
NEXT f:ERASE shadow:COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB request(wid%,hi%,bt%) STATIC [RETURN]
SHARED add&,st$,res&,body$,yes$,no$,offs%
[RETURN]
opt&=2^1+2^16:offs%=0 [RETURN]
req&=AllocRemember&(0,400,opt&) [RETURN]
IF req&=0 THEN ERROR 7 [RETURN]
add&=req& [RETURN]
t1&=add& [RETURN]
FOR loop2=0 TO bt%-1 [RETURN]
st$=body$(loop2) [RETURN]
makeheader add&,st$,1,5,offs%+3 [RETURN]
offs%=offs%+8 [RETURN]
NEXT loop2 [RETURN]
st$=body$(bt%) [RETURN]
makeheader add&,st$,0,5,offs%+3 [RETURN]
st$=yes$ [RETURN]
t2&=add& [RETURN]
makeheader add&,st$,0,5,3 [RETURN]
st$=no$ [RETURN]
t3&=add& [RETURN]
makeheader add&,st$,0,5,3 [RETURN]
res&=autorequest&(WINDOW(7),t1&,t2&,t3&,0,0,
wid%,hi%) [RETURN]
CALL freeremember(0,-1) [RETURN]
END SUB [RETURN]
SUB makeheader(ptr&,text$,md%,le%,te%)
STATIC [RETURN]
SHARED add& [RETURN]
text$=text$+CHR$(0):POKE ptr&,1:POKE
ptr&+1,0 [RETURN]
POKE ptr&+2,2:POKEW ptr&+4,le%:POKEW
ptr&+6,te% [RETURN]
POKEL ptr&+8,0:POKEL ptr&+12,SADD(text$)
[RETURN]
IF md%=0 THEN [RETURN]
POKEL ptr&+16,0 [RETURN]
ELSE [RETURN]
POKEL ptr&+16,ptr&+20 [RETURN]
END IF [RETURN]
add&=ptr&+20 [RETURN]
END SUB [RETURN]
SUB display (text$) STATIC [RETURN]
CALL
text&(WINDOW(8),SADD(text$),LEN(text$))
[RETURN]
END SUB [RETURN]
SUB texting (px2%,py2%,tekst$) STATIC
[RETURN]
e&=Move&(WINDOW(8),px2%,py2%) [RETURN]
CALL
text&(WINDOW(8),SADD(tekst$),LEN(tekst$))
[RETURN]
END SUB [RETURN]
SUB style(style%) STATIC [RETURN]
bits&=Asksoftstyle&(WINDOW(8)) [RETURN]
newstyle&=Setsoftstyle&(WINDOW(8),style%,bit
s&) [RETURN]
```

```
END SUB [RETURN]
SUB getdir (diri$) STATIC [RETURN]
SHARED counter%,maxidir%,fault% [RETURN]
diri$=diri$+CHR$(0):counter%=0:bytes&=252:fa
ult%=0:er&=Lock&(SADD(diri$),-2) [RETURN]
IF er&=0 THEN fault%=1:EXIT SUB [RETURN]
opt&=2^1+2^16:info&=Allocmem&(bytes&,opt&):s
uc&=Examine&(er&,info&) [RETURN]
IF suc&=0 THEN EXIT SUB [RETURN]
again: [RETURN]
dirname&=info&+8 [RETURN]
FOR search=0 TO 29 [RETURN]
check=PEEK(dirname&+search) [RETURN]
IF check<>0 THEN check$=check$+CHR$(check)
ELSE search=29 [RETURN]
NEXT search [RETURN]
dirname$(counter%)=check$:check$="":type&=PE
EKL(info&+120):dirtype$="" [RETURN]
IF type&>-1 AND counter%>0 THEN dirtype$="
(Dir)" [RETURN]
dirname$(counter%)=dirname$(counter%)+dirtyp
e$:suc&=Exnext&(er&,info&):IF suc&=0 THEN
finish [RETURN]
counter%=counter%+1:IF counter%<=maxidir%
THEN GOTO again [RETURN]
finish: [RETURN]
CALL freemem(info&,bytes&):CALL
Unlock(er&) [RETURN]
END SUB [RETURN]
SUB dir (x%,y%,naam$,path$,extra$) STATIC
[RETURN]
SHARED
dirchoice,counter%,x2%,y2%,place,inv$,inv,fa
ult%,drawer$,file$,program$ [RETURN]
x2%=x%:y2%=y%:until%=1:POKE 12574721&,254
[RETURN]
IF odraw$<>path$ THEN [RETURN]
IF keer=1 THEN ERASE position9 [RETURN]
keer=1:getdir
path$:path$=LEFT$(path$,LEN(path$)-1):drawer
$=path$ [RETURN]
DIM position9(7,4):RESTORE dirdata:place=1
[RETURN]
FOR f=1 TO 7:FOR g=1 TO 4:READ
position9(f,g):NEXT g:NEXT f [RETURN]
END IF [RETURN]
GET (x%,y%)-(x%+200,y%+130),dirblock%
[RETURN]
LINE (x%,y%)-(x%+200,y%+130),0,bf:LINE
(x%,y%)-(x%+200,y%+11),1,bf:COLOR 0,1
[RETURN]
texting x%+8,y%+9,naam$:LINE
(x%,y%+13)-(x%+183,y%+77),1,bf [RETURN]
LINE
(x%+186,y%+13)-(x%+200,y%+25),1,bf:LINE
(x%+186,y%+27)-(x%+200,y%+63),1,bf [RETURN]
LINE (x%+186,y%+65)-(x%+200,y%+77),1,bf
[RETURN]
AREA (x%+188,y%+23):AREA STEP(10,0):AREA
STEP(-5,-8):AREA STEP(-5,8):AREAFILL
[RETURN]
AREA (x%+188,y%+67):AREA STEP(10,0):AREA
STEP(-5,8):AREA STEP(-5,-8):AREAFILL
[RETURN]
LINE
(x%,y%+79)-(x%+200,y%+130),1,bf:texting
x%+4,y%+87,"Drawer":texting
x%+4,y%+97,"File" [RETURN]
LINE
(x%+60,y%+80)-(x%+197,y%+88),0,bf:LINE
(x%+60,y%+90)-(x%+197,y%+98),0,bf [RETURN]
LINE
(x%+4,y%+102)-(x%+65,y%+113),0,bf:LINE
(x%+70,y%+102)-(x%+131,y%+113),0,bf
[RETURN]
LINE
(x%+136,y%+102)-(x%+196,y%+113),0,bf:LINE
(x%+36,y%+116)-(x%+97,y%+127),0,bf [RETURN]
LINE
(x%+103,y%+116)-(x%+168,y%+127),0,bf:COLOR
1,0:texting x%+19,y%+110,"DF0:" [RETURN]
texting x%+87,y%+110,"DF1:":texting
x%+65,y%+87,RIGHT$(drawer$,16) [RETURN]
IF extra$="" OR RIGHT$(extra$,1)<>":" THEN
extra$="RAM:" [RETURN]
texting x%+152,y%+110,extra$:texting
x%+44,y%+124,"Accept":texting
x%+113,y%+124,"Cancel":fresh [RETURN]
dirmouseloop: WHILE MOUSE(0)>=0:WEND
[RETURN]
px=MOUSE(1):py=MOUSE(2) [RETURN]
clickpart=0 [RETURN]
IF px>x%-1 AND px<x%+184 AND py>y%+12 AND
py<y%+78 THEN [RETURN]
clickpart=1 [RETURN]
ELSEIF px>x%+185 AND px<x%+201 AND
py>y%+12 AND py<y%+78 THEN [RETURN]
clickpart=2 [RETURN]
ELSEIF px>x%-1 AND px<x%+201 AND py>y%+78
AND py<y%+131 THEN [RETURN]
clickpart=3 [RETURN]
END IF [RETURN]
IF clickpart=0 THEN [RETURN]
GOTO dirmouseloop [RETURN]
ELSEIF clickpart=1 THEN [RETURN]
oldunt%=until% [RETURN]
FOR f=1 TO 8 [RETURN]
IF py>y%+12+((f-1)*8) AND py<y%+13+(f*8)
THEN until%=f:f=8 [RETURN]
NEXT f [RETURN]
difference%=until%-oldunt% [RETURN]
place=place+difference% [RETURN]
COLOR 0,1:texting
x%+175,y%+12+(8*oldunt%)," ":COLOR 1,0
[RETURN]
IF RIGHT$(dirname$(place),5)="(Dir)" THEN
[RETURN]
IF RIGHT$(drawer$,1)<>":" THEN
drawer$=drawer$+"/" [RETURN]
path2$=dirname$(place):drawer$=drawer$+LEFT$
(path2$,LEN(path2$)-6) [RETURN]
verwerk x%+65,y%+87,drawer$,16 [RETURN]
getdir drawer$:place=1:until%=1:fresh
[RETURN]
drawer$=LEFT$(drawer$,LEN(drawer$)-1)
[RETURN]
ELSEIF file$<>dirname$(place) THEN [RETURN]
file$=dirname$(place):app$="" [RETURN]
verwerk x%+65,y%+97,RIGHT$(file$,16),16
[RETURN]
END IF [RETURN]
COLOR 0,1:texting
x%+175,y%+12+(8*until%),"-":COLOR 1,0
[RETURN]
ELSEIF clickpart=2 AND counter%>0 THEN
[RETURN]
COLOR 0,1:texting
x%+175,y%+12+(8*until%)," " [RETURN]
IF py<y%+26 THEN [RETURN]
IF place>1 THEN [RETURN]
place=place-1:until%=until%-1 [RETURN]
IF until%<1 THEN [RETURN]
```

```
place=place-7:IF place<1 THEN place=1
[RETURN]
until%=1:fresh:place=place+7 [RETURN]
COLOR 0,1:texting x%+175,y%+20,"
":until%=8:texting x%+175,y%+76,"-"
[RETURN]
END IF [RETURN]
END IF [RETURN]
ELSEIF py>y%+65 THEN [RETURN]
IF place<counter% THEN place=place+1
[RETURN]
until%=until%+1:IF until%>8 THEN
:fresh:until%=1 [RETURN]
ELSE [RETURN]
COLOR 1,0 [RETURN]
IF RIGHT$(dirname$(place),5)="(Dir)" THEN
[RETURN]
IF RIGHT$(drawer$,1)<>":" THEN
drawer$=drawer$+"/" [RETURN]
path2$=dirname$(place):drawer$=drawer$+LEFT$
(path2$,LEN(path2$)-6) [RETURN]
verwerk
x%+65,y%+87,RIGHT$(drawer$,16),16:getdir
drawer$:place=1 [RETURN]
until%=1:fresh:drawer$=LEFT$(drawer$,LEN(dra
wer$)-1) [RETURN]
ELSEIF file$<>dirname$(place) THEN [RETURN]
file$=dirname$(place):verwerk
x%+65,y%+97,RIGHT$(file$,16),16 [RETURN]
END IF [RETURN]
END IF [RETURN]
COLOR 0,1:texting
x%+175,y%+12+(8*until%),"-":COLOR 1,0
[RETURN]
ELSE [RETURN]
dirchoice=0 [RETURN]
FOR f=1 TO 7 [RETURN]
xway=0:yway=0 [RETURN]
IF px>x%+position9(f,1) AND
px<x%+position9(f,3) THEN xway=1 [RETURN]
IF py>y%+position9(f,2) AND
py<y%+position9(f,4) THEN yway=1 [RETURN]
IF xway=1 AND yway=1 THEN dirchoice=f:f=7
[RETURN]
NEXT f [RETURN]
IF dirchoice=6 OR dirchoice=7 THEN [RETURN]
PUT (x%,y%),dirblock%,PSET:POKE
12574721&,252:odraw$=path$ [RETURN]
COLOR 1,0:IF RIGHT$(drawer$,1)<>":" AND
RIGHT$(drawer$,1)<>"/" THEN p$="/" ELSE
p$="" [RETURN]
program$=drawer$+p$+file$:EXIT SUB [RETURN]
ELSEIF dirchoice=1 THEN [RETURN]
ans=LEN(drawer$)+1:ed
drawer$,16,255,x%+65,y%+87:path$=inv$:existi
ng=0:er&=Lock&(SADD(path$),-2) [RETURN]
IF UCASE$(drawer$)<>UCASE$(path$) AND
er&>0 THEN [RETURN]
drawer$=path$:getdir path$ [RETURN]
place=1:until%=1:fresh [RETURN]
END IF [RETURN]
verwerk x%+65,y%+87,RIGHT$(drawer$,16),16
[RETURN]
ELSEIF dirchoice=2 THEN [RETURN]
ed file$,16,255,x%+65,y%+97 [RETURN]
IF UCASE$(inv$)<>UCASE$(file$) THEN
[RETURN]
file$=inv$ [RETURN]
END IF [RETURN]
verwerk x%+65,y%+97,RIGHT$(file$,16),16
[RETURN]
ELSEIF dirchoice=3 THEN [RETURN]
path$="DF0:":getdir path$:drawer$="DF0:"
[RETURN]
place=1:until%=1:fresh:verwerk
x%+65,y%+87,drawer$,16 [RETURN]
ELSEIF dirchoice=4 THEN [RETURN]
path$="DF1:":getdir path$:drawer$="DF1:"
[RETURN]
place=1:until%=1:fresh:verwerk
x%+65,y%+87,drawer$,16 [RETURN]
ELSEIF dirchoice=5 THEN [RETURN]
path$=extra$:getdir path$:drawer$=extra$
[RETURN]
place=1:until%=1:fresh:verwerk
x%+65,y%+87,drawer$,16 [RETURN]
END IF [RETURN]
END IF [RETURN]
GOTO dirmouseloop [RETURN]
END SUB [RETURN]
dirdata: [RETURN]
DATA
59,79,198,89,59,89,198,99,3,101,66,114,69,10
1,132,114,135,101,197,114,35,115,98,128,102,
115,169,128 [RETURN]
SUB fresh STATIC [RETURN]
SHARED place,x2%,y2%,counter% [RETURN]
COLOR 0,1 [RETURN]
h%=0 [RETURN]
FOR f%=place TO place+7:h%=h%+1:app$=""
[RETURN]
e&=Move&(WINDOW(8),x2%+4,y2%+12+(8*h%))
[RETURN]
IF LEN(dirname$(f%))<21 THEN
app$=STRING$(21-LEN(dirname$(f%))," ")
[RETURN]
IF NOT(f%>counter%) THEN display
MID$(dirname$(f%),1,21)+app$ ELSE display
STRING$(21," ") [RETURN]
e&=Move&(WINDOW(8),x2%+175,y2%+12+(8*h%))
[RETURN]
IF f%<>place THEN display " " ELSE
:display "-" [RETURN]
NEXT f% [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB ed (text$,length%,max%,xp%,yp%) STATIC
[RETURN]
SHARED inv$,inv [RETURN]
inv$=text$:key$="":key=0 [RETURN]
texting xp%,yp%,STRING$(length%," ")
[RETURN]
texting xp%,yp%,RIGHT$(inv$,length%-1)+"_"
[RETURN]
WHILE NOT (key=13 OR key=27 OR
LEN(inv$)=>max%) [RETURN]
key$="":key=-1 [RETURN]
WHILE key$="":key$=INKEY$:WEND [RETURN]
key=ASC(key$) [RETURN]
IF key=8 OR key=127 THEN [RETURN]
IF LEN(inv$)=0 THEN slaoverlabel: [RETURN]
inv$=LEFT$(inv$,LEN(inv$)-1):verwerk
xp%,yp%,RIGHT$(inv$,length%-1)+"_",length%
[RETURN]
slaoverlabel: [RETURN]
ELSE [RETURN]
inv$=inv$+key$ [RETURN]
IF key<>13 THEN texting
xp%,yp%,RIGHT$(inv$,length%-1)+"_" [RETURN]
END IF [RETURN]
WEND [RETURN]
```

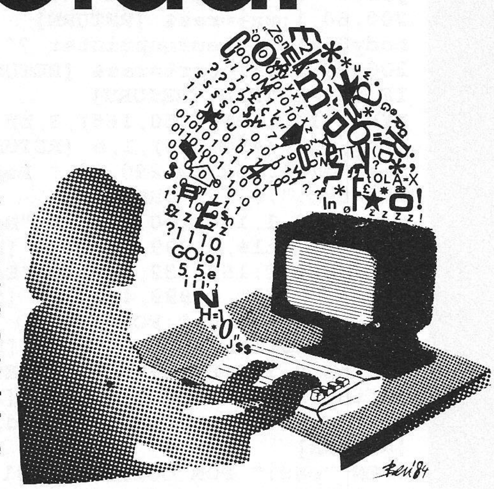
```
inv$=LEFT$(inv$,LEN(inv$)-1):IF key=27
THEN inv$=text$ [RETURN]
inv=VAL(inv$) [RETURN]
END SUB [RETURN]
SUB verwerk (px2%,py2%,tekst$,length%)
STATIC [RETURN]
tekst2$="":IF LEN(tekst$)<length% THEN
tekst2$=STRING$(length%-LEN(tekst$)," ")
[RETURN]
texting
px2%,py2%,RIGHT$(tekst$,length%)+tekst2$
[RETURN]
END SUB [RETURN]
SUB create6 (num%) STATIC [RETURN]
DIM SHARED
position6(num%,3),text6$(num%),scale6(num%,3
),colour6(num%,3),varia6(num%),path6$(num%)
[RETURN]
END SUB [RETURN]
SUB define6
(num%,px%,py%,lenx%,tekst$,sch1%,sch2%,vari%
,sch3%,col1%,col2%,col3%) STATIC [RETURN]
position6(num%,1)=px%:position6(num%,2)=py%
[RETURN]
position6(num%,3)=lenx%:text6$(num%)=tekst$
[RETURN]
scale6(num%,1)=sch1%:scale6(num%,2)=sch2%
[RETURN]
scale6(num%,3)=sch3%:colour6(num%,1)=col1%
[RETURN]
colour6(num%,2)=col2%:colour6(num%,3)=col3%
[RETURN]
varia6(num%)=vari%:path6$(num%)=STRING$(LEN(
MID$(STR$(sch2%),2)),"#") [RETURN]
END SUB [RETURN]
SUB show6 (num%) STATIC [RETURN]
FOR f=1 TO num% [RETURN]
px%=position6(f,1)-((LEN(text6$(f))+1)*8):py
%=position6(f,2) [RETURN]
COLOR colour6(f,2):texting
px%,py%+8,text6$(f):px%=position6(f,1)
[RETURN]
LINE
(px%,py%)-(px%+position6(f,3),py%+10),colour
6(f,1),bf [RETURN]
LINE
(px%,py%)-(px%+position6(f,3),py%+10),colour
6(f,2),b [RETURN]
IF scale6(f,3)=1 THEN [RETURN]
pox%=px%+position6(f,3)+8 [RETURN]
e&=Move&(WINDOW(8),pox%,py%+8) [RETURN]
PRINT USING path6$(f);varia6(f) [RETURN]
END IF [RETURN]
ps=(position6(f,3)/(scale6(f,2)-scale6(f,1))
)*(varia6(f)-scale6(f,1)) [RETURN]
LINE
(px%+ps,py%+1)-(px%+ps,py%+9),colour6(f,3)
[RETURN]
NEXT f [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB control6 (num%) STATIC [RETURN]
SHARED choicemade [RETURN]
x=MOUSE(1):y=MOUSE(2) [RETURN]
choicemade=0 [RETURN]
FOR f=1 TO num% [RETURN]
xway=0:yway=0 [RETURN]
IF x>position6(f,1)-1 AND
x<position6(f,1)+position6(f,3)+1 THEN
xway=1 [RETURN]
IF y>position6(f,2)-2 AND
y<position6(f,2)+10 THEN yway=1 [RETURN]
IF xway=1 AND yway=1 THEN
choicemade=f:f=num% [RETURN]
NEXT f [RETURN]
IF choicemade=0 THEN EXIT SUB [RETURN]
f=choicemade:ps=(position6(f,3)/(scale6(f,2)
-scale6(f,1)))*(varia6(f)-scale6(f,1))
[RETURN]
WHILE NOT (MOUSE(0)>=0 OR
MOUSE(1)<position6(f,1) OR
MOUSE(1)>position6(f,1)+position6(f,3)+1
OR MOUSE(2)<position6(f,2) OR
MOUSE(2)>position6(f,2)+10) [RETURN]
x=MOUSE(1) [RETURN]
IF varia6(f)<>scale6(f,1) AND
varia6(f)<>scale6(f,2) THEN LINE
(position6(f,1)+ps,position6(f,2)+1)-(positi
on6(f,1)+ps,position6(f,2)+9),colour6(f,1)
[RETURN]
ps=x-position6(f,1) [RETURN]
LINE
(position6(f,1)+ps,position6(f,2)+1)-(positi
on6(f,1)+ps,position6(f,2)+9),colour6(f,3)
[RETURN]
a=position6(f,3)/(scale6(f,2)-scale6(f,1)):v
aria6(f)=(ps/a)+scale6(f,1) [RETURN]
IF scale6(f,3)=1 THEN [RETURN]
pox%=position6(f,1)+position6(f,3)+8:py%=pos
ition6(f,2)+8 [RETURN]
e&=Move&(WINDOW(8),pox%,py%) [RETURN]
PRINT USING path6$(f);varia6(f) [RETURN]
END IF [RETURN]
WEND [RETURN]
COLOR 1,0 [RETURN]
END SUB [RETURN]
SUB create (n%) STATIC [RETURN]
DIM SHARED controllen(n%,5):FOR f=1 TO
n%:controllen(n%,0)=-1:NEXT f [RETURN]
END SUB [RETURN]
SUB define (n%,x%,y%,xl%,yl%,max%) STATIC
[RETURN]
controllen(n%,1)=x%:controllen(n%,2)=y%:cont
rollen(n%,0)=0 [RETURN]
controllen(n%,3)=xl%:controllen(n%,4)=yl%:co
ntrollen(n%,5)=max% [RETURN]
END SUB [RETURN]
SUB master (n%) STATIC [RETURN]
SHARED choicemade [RETURN]
ret%=0 [RETURN]
masterloop: WHILE MOUSE(0)>=0:WEND [RETURN]
px=MOUSE(1):py=MOUSE(2):choicemade=0
[RETURN]
FOR f=1 TO n%:clicked%=0 [RETURN]
IF controllen(f,0)=-1 OR f=9 THEN ml1
[RETURN]
IF px>controllen(f,1)-1 AND
px<controllen(f,1)+controllen(f,3)+1 AND
py>controllen(f,2)-1 AND
py<controllen(f,2)+controllen(f,4)+1 THEN
clicked%=1 [RETURN]
IF clicked%=1 THEN ON f GOSUB
con1,con2,con3,con4,con5,con6,con7,con8,0,co
n10,con11 [RETURN]
IF (f=2 OR f=3) AND choicemade<>0 THEN
ret%=1 [RETURN]
IF choicemade<>0 THEN f=n% [RETURN]
ml1: NEXT f [RETURN]
IF ret%=0 THEN masterloop [RETURN]
EXIT SUB [RETURN]
```

```
con1: a%=controllen(1,5):Control1
a%:RETURN [RETURN]
con2: a%=controllen(2,5):control2
a%:RETURN [RETURN]
con3: a%=controllen(3,5):Control3
a%:RETURN [RETURN]
con4: a%=controllen(4,5):Control4
a%:RETURN [RETURN]
con5: a%=controllen(5,5):Control5
a%:RETURN [RETURN]
con6: a%=controllen(6,5):control6
a%:RETURN [RETURN]
con7: a%=controllen(7,5):Control7
a%:RETURN [RETURN]
con8: a%=controllen(8,5):control8
a%:RETURN [RETURN]
con10: a%=controllen(10,5):control10
a%:RETURN [RETURN]
con11: a%=controllen(11,5):control11
a%:RETURN [RETURN]
END SUB [RETURN]
a: [RETURN]
SUB dorest STATIC [RETURN]
WHILE NOT EOF(1) [RETURN]
LINE INPUT#1,a$ [RETURN]
PRINT#2,a$ [RETURN]
WEND [RETURN]
CLOSE#2:CLOSE#1 [RETURN]
END SUB [RETURN]
SUB plaats STATIC [RETURN]
SHARED
lettertyp%,stijl%,onderlijn%,vet%,cursief%,t
kst$,endje [RETURN]
stijl%=position11(1,5) [RETURN]
IF stand8(1)=1 THEN lettertyp%=80 ELSE
lettertyp%=77 [RETURN]
IF stand8(3)=1 THEN onderlijn%=49 ELSE
onderlijn%=48 [RETURN]
IF stand8(5)=1 THEN vet%=71 ELSE vet%=72
[RETURN]
IF stand8(7)=1 THEN cursief%=52 ELSE
cursief%=53 [RETURN]
PRINT#2,STR$(lettertyp%)+";" [RETURN]
PRINT#2,STR$(stijl%)+";" [RETURN]
PRINT#2,STR$(onderlijn%)+";" [RETURN]
PRINT#2,STR$(vet%)+";" [RETURN]
PRINT#2,STR$(cursief%)+";" [RETURN]
PRINT#2,tkst$+";" [RETURN]
IF endje=1 THEN a$="END;" ELSE a$=";"
[RETURN]
PRINT#2,a$ [RETURN]
END SUB [RETURN]
uitvoer: [RETURN]
CLS:titel "FORM, geschreven door Swiddy de
Louw.              PRINT-OUT","FORM"
[RETURN]
COLOR 2,0:AREA (105,112):AREA STEP
(190,-67):AREA STEP (190,67):AREA STEP
(-190,67):AREA STEP (-190,-67):AREAFILL
[RETURN]
COLOR 1,0:LINE (105,112)-(295,45),1:LINE
-STEP(190,67):LINE -STEP(-190,67):LINE
-STEP(-190,-67) [RETURN]
define2 1,140,60,150,50,"
PRINTER",3,1,2 [RETURN]
define2 2,300,60,150,50,"
BEELDSCHERM",3,1,2 [RETURN]
define2 3,140,115,150,50," PRINTER
VERKORT",3,1,2 [RETURN]
define2 4,300,115,150,50,"
HOOFDMENU",3,1,2 [RETURN]
show2 4:choicemade=0 [RETURN]
WHILE choicemade=0 [RETURN]
WHILE MOUSE(0)>=0:WEND [RETURN]
control2 4 [RETURN]
WEND [RETURN]
IF choicemade=4 THEN [RETURN]
RETURN [RETURN]
ELSEIF choicemade=2 THEN [RETURN]
CLS:style 6:texting 20,16,"Uitvoer naar
beeldscherm.":style 0 [RETURN]
define2 1,19,210,571,16,"
STOP",2,1,1 [RETURN]
show2 1:style 2:texting 15,28,"Nummer
Categorie          Naam programma
Kleur":style 0 [RETURN]
OPEN bestand$ FOR INPUT AS
#1:zolang=0:teller=5 [RETURN]
WHILE zolang=0 [RETURN]
LINE INPUT#1,num$:IF UCASE$(num$)="END"
THEN zolang=1:teller=23:GOTO p [RETURN]
LINE INPUT#1,naam$ [RETURN]
LINE INPUT#1,cate$:LINE
INPUT#1,a$:INPUT#1,b$ [RETURN]
kl=VAL(a$):num$=LEFT$(num$,LEN(num$)-1)
[RETURN]
naam$=LEFT$(naam$,LEN(naam$)-1):cate$=LEFT$(
cate$,LEN(cate$)-1) [RETURN]
LOCATE teller,4:PRINT num$;"
";LEFT$(cate$,17);TAB(30);LEFT$(naam$,35);TA
B(69);kl [RETURN]
d=MOUSE(0) [RETURN]
IF d<0 THEN [RETURN]
control2 1 [RETURN]
IF choicemade=1 THEN zolang=1:GOTO
eindeloep [RETURN]
END IF [RETURN]
p: teller=teller+1 [RETURN]
IF teller>23 THEN [RETURN]
teller=5 [RETURN]
COLOR 1,2:texting 195,222,"Druk op een
(muis) toets." [RETURN]
SLEEP:SLEEP:COLOR 1,0 [RETURN]
LINE (18,39)-(600,202),0,bf:show2 1
[RETURN]
END IF     [RETURN]
eindeloep: [RETURN]
WEND [RETURN]
CLOSE#1 [RETURN]
ELSEIF choicemade=3 THEN [RETURN]
body$(1)=" Is printer gereed ?" [RETURN]
yes$=" Ja ":no$=" Nee ":request 200,60,1
[RETURN]
IF res&=1 THEN [RETURN]
OPEN "par:" FOR OUTPUT AS #2 [RETURN]
OPEN bestand$ FOR INPUT AS #1 [RETURN]
zolang=0:PRINT#2,CHR$(27);CHR$(71);"NUMMER
CATEGORIE";STRING$(13,32);"PROGRAMMANAAM";ST
RING$(29,32);"KLEUR";CHR$(27);CHR$(72):PRINT
#2,CHR$(13) [RETURN]
WHILE zolang=0 [RETURN]
LINE INPUT#1,num$:IF UCASE$(num$)="END"
THEN zolang=1:GOTO o [RETURN]
LINE INPUT#1,naam$ [RETURN]
LINE INPUT#1,cate$:LINE
INPUT#1,a$:INPUT#1,b$ [RETURN]
kl=VAL(a$):num$=LEFT$(num$,LEN(num$)-1)
[RETURN]
naam$=LEFT$(naam$,LEN(naam$)-1):cate$=LEFT$(
cate$,LEN(cate$)-1) [RETURN]
```

```
PRINT#2," ";num$;"
";LEFT$(cate$,20);STRING$(22-LEN(cate$),32);
naam$;STRING$(43-LEN(naam$),32);kl [RETURN]
o: [RETURN]
WEND [RETURN]
CLOSE#1:CLOSE#2 [RETURN]
END IF [RETURN]
ELSEIF choicemade=1 THEN [RETURN]
body$(1)=" Is printer gereed ?" [RETURN]
yes$=" Ja ":no$=" Nee ":request
200,60,1:xz%=res& [RETURN]
body$(1)=" Kleurenprinter ?":request
200,60,1:printer%=res& [RETURN]
IF xz%=1 THEN [RETURN]
LINE (140,60)-(450,165),3,bf:LINE
(140,60)-(450,165),1,b [RETURN]
define2 1,150,69,290,20," Begin met
printen.",0,1,1 [RETURN]
define10 1,150,100,290,20,"Beginnummer van
label.",stl%,0,999,4,0,1,1 [RETURN]
define10 2,150,132,290,20,"Eindnummer van
label.",edl%,0,999,4,0,1,1 [RETURN]
show2 1:show10 2:FOR f=1 TO
11:controllen(f,0)=-1:NEXT [RETURN]
define 2,150,69,290,20,1 [RETURN]
define 10,150,100,290,52,2 [RETURN]
master 10:stl%=value10(1):edl%=value10(2)
[RETURN]
OPEN "par:" FOR OUTPUT AS #1 [RETURN]
cpr%=INT(breedte%/3):lth%=INT(hoogte%/4):zol
ang=0:PRINT#1,CHR$(27);CHR$(108);CHR$(lmarge
%); [RETURN]
PRINT#1,CHR$(27);CHR$(120);CHR$(48+nlqofdraf
t%); [RETURN]
OPEN bestand$ FOR INPUT AS #2 [RETURN]
w=0 [RETURN]
WHILE zolang=0          [RETURN]
w=w+1 [RETURN]
LINE INPUT#2,num$:IF UCASE$(num$)="END"
THEN zolang=1:GOTO u [RETURN]
LINE INPUT#2,naam$ [RETURN]
LINE INPUT#2,cate$:LINE
INPUT#2,a$:INPUT#2,b$ [RETURN]
kl=VAL(a$):num$=LEFT$(num$,LEN(num$)-1)
[RETURN]
naam$=LEFT$(naam$,LEN(naam$)-1):cate$=LEFT$(
cate$,LEN(cate$)-1):z2=0 [RETURN]
IF w>=stl% AND w<=edl% THEN     [RETURN]
OPEN bron$ FOR INPUT AS #3:teller=0
[RETURN]
WHILE z2=0 [RETURN]
LINE INPUT#3,a$:lt%=VAL(a$) [RETURN]
LINE INPUT#3,a$:styl%=VAL(a$) [RETURN]
LINE INPUT#3,a$:onder%=VAL(a$) [RETURN]
LINE INPUT#3,a$:vet%=VAL(a$) [RETURN]
LINE INPUT#3,a$:cursief%=VAL(a$) [RETURN]
LINE INPUT#3,a$:tekst$=LEFT$(a$,LEN(a$)-1)
[RETURN]
LINE INPUT#3,a$:IF UCASE$(a$)="END;" THEN
z2=1 [RETURN]
'instellen van printer
(vet,onderlijnen,lettertype, etc.) [RETURN]
PRINT#1,CHR$(27);CHR$(lt%);CHR$(27);CHR$(45)
;CHR$(onder%); [RETURN]
PRINT#1,CHR$(27);CHR$(vet%);CHR$(27);CHR$(cu
rsief%); [RETURN]
IF printer%=1 THEN
PRINT#1,CHR$(27);CHR$(114);CHR$(kl);
[RETURN]
IF styl%=1 THEN [RETURN]
m%=cpr%:PRINT#1,CHR$(27);CHR$(87);CHR$(48);
[RETURN]
PRINT#1,CHR$(18);CHR$(27);CHR$(70);
[RETURN]
ELSEIF styl%=2 THEN [RETURN]
m%=cpr%:PRINT#1,CHR$(27);CHR$(87);CHR$(48);
[RETURN]
PRINT#1,CHR$(18);CHR$(27);CHR$(69);
[RETURN]
ELSEIF styl%=3 THEN [RETURN]
m%=1.71*cpr%:PRINT#1,CHR$(27);CHR$(87);CHR$(
48); [RETURN]
PRINT#1,CHR$(15);CHR$(27);CHR$(70);
[RETURN]
ELSEIF styl%=4 THEN [RETURN]
m%=cpr%/1.67:PRINT#1,CHR$(27);CHR$(87);CHR$(
49); [RETURN]
PRINT#1,CHR$(18);CHR$(27);CHR$(70);
[RETURN]
END IF [RETURN]
a=INSTR(UCASE$(tekst$),"[NAAM]") [RETURN]
b=INSTR(UCASE$(tekst$),"[CATEGORIE]")
[RETURN]
c=INSTR(UCASE$(tekst$),"[NUMMER]") [RETURN]
IF a>0 THEN
tekst$=LEFT$(tekst$,a-1)+naam$+MID$(tekst$,a
+6) [RETURN]
IF b>0 THEN
tekst$=LEFT$(tekst$,b-1)+cate$+MID$(tekst$,b
+11) [RETURN]
IF c>0 THEN
tekst$=LEFT$(tekst$,c-1)+num$+MID$(tekst$,c+
8) [RETURN]
IF LEN(tekst$)>m% THEN [RETURN]
BEEP [RETURN]
tekst$=LEFT$(tekst$,m%) [RETURN]
END IF [RETURN]
PRINT#1,tekst$ [RETURN]
teller=teller+1:IF teller>lth% THEN z2=1
[RETURN]
WEND [RETURN]
u: [RETURN]
CLOSE#3 [RETURN]
FOR f=1 TO
lth%-teller:PRINT#1,CHR$(13):NEXT [RETURN]
END IF [RETURN]
WEND [RETURN]
PRINT#1,CHR$(27);CHR$(80);CHR$(27);CHR$(45);
CHR$(48); [RETURN]
PRINT#1,CHR$(27);CHR$(72);CHR$(27);CHR$(53);
CHR$(27);CHR$(114);CHR$(0); [RETURN]
CLOSE#1:CLOSE#2 [RETURN]
END IF [RETURN]
END IF [RETURN]
GOTO uitvoer [RETURN]
RETURN [RETURN]
```

Een inleiding tot de wonderbaarlijke wereld van de 68000-serie, de copper, blitter, CIA's, en nog veel meer. Beleef mee hoe de Amiga haar geheugen indeelt, bitjes manipuleert, programma's laat communiceren, en met schermen en vensters goochelt. Ervaar de zinderende snelheid van...

Amiga machinetaal



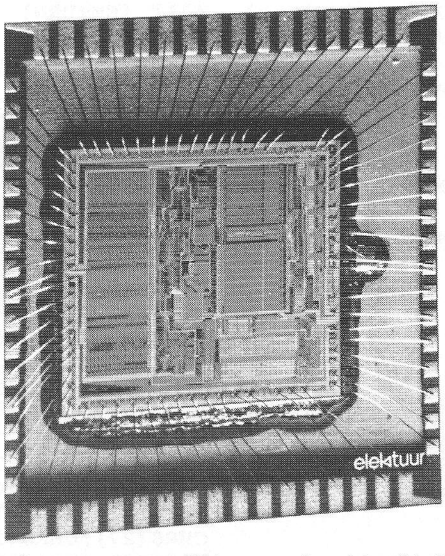
Natuurlijk, zo geweldig en flitsend als hierboven geschetst wordt, zal 't in het begin niet gaan. Maar je allereerste assemblerprogramma te zien werken... dat geeft toch wel een bepaalde 'kick'. We kunnen dan terecht van een KICKSTART praten! Voordat de bitjes en blitjes ons om de oren vliegen, moeten we toch eerst even wat feiten op een rijtje zetten. Want als er straks sprake zal zijn van registers, adressen, statusbits etc. is het wel zo leuk om te weten waar we het dan over hebben. Ook voor diegenen die vroeger al met 6502 machinetaal bezig zijn geweest, op de Commodore 64 of 128 waarschijnlijk, zullen een aantal begrippen en principes tegenkomen die ze al kennen. Toch is het zinvol om er even bij te blijven, want een 68000 processor is toch weer net ietsje anders. Ongeveer twee keer de helft namelijk. Onder 'feiten' verstaan we vooral de opbouw van de 68000 processor, voortaan zullen wij deze processor als 68000 aanduiden. Verder gaan we als eerste bekijken, opdat je een idee krijgt van de vorm en functie van registers.

68000 Intern

Zoals iedereen wel begrijpen zal, moet een 68000 gegevens hebben waarmee gewerkt kan worden; bv. 2 getallen voor een vermenigvuldiging. Deze gegevens kunnen uit 2 plaatsen komen: de interne registers van de 68000, en het RAM. Er zijn 20 registers in de 68000 aanwezig:
D0...D7 : 8 dataregisters, deze zijn 4 bytes groot, en kunnen zodoende 32 bits bevatten. Ze kunnen rekenen met bytes, words (2 bytes), en longwords (4 bytes). **A0...A7 :** 8 adresregisters deze zijn ook 4 bytes groot, maar kunnen slechts voor word- en longwordoperaties gebruikt worden. Verwijzingen naar het RAM-geheugen kunnen alleen via deze registers gebeuren. Register A7 heeft een speciale functie; n.l. als stackpointer. **PC :** program counter hierin staat het adres van de instructie die op dat moment door de processor uitgevoerd wordt. **USP en SSP:** stackpointers **SR :** status register dit bevat bits, die de status van de processor weergeven. Deze status kan veranderd worden naar aanleiding van het resultaat van een operatie. De inhoud is als volgt:

```
Systeem byte          User byte
-------------------------------
Bit15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
      T    S           I I I       X N Z V C
```

De user byte bevat 5 bits die de conditie van de processor weergeven na een operatie, zoals een rekenfunctie



of een vergelijkingsopdracht. Met deze bits kan de uitkomst van zo'n operatie gebruikt worden voor een sprongopdracht, zoiets als 'sla 20 bytes over als het resultaat nul is'.
Bit 0: carry bit C bij een rekenfunctie is een overloop ('een onthouden') voorgekomen. **Bit 1:** overflow bit V het resultaat van een rekenfunctie is te groot voor het antwoord-register of -adres. **Bit 2:** zero bit N het resultaat is nul. **Bit 3:** negatief bit het resultaat van een operatie is negatief. **Bit 4:** extend bit X heeft ongeveer dezelfde functie als carry-bit C, maar wordt door minder instructies gebruikt.
Op de exacte functie van de systeembyte komen we later terug; hier volstaan we voorlopig met een korte beschrijving. **Bit 8/9/10:** interrupt masker de combinatie van deze 3 bits geeft aan van welke bron een interrupt afkomstig is. **Bit 13:** supervisor bit heeft de waarde '1' als de processor in supervisor-status werkt, de waarde '0' voor user-status. **Bit 15:** trace bit deze heeft de waarde '1' als de 68000 in trace-mode werkt.

RAM

Het geheugen buiten de processor bestaat uit RAM, waarmee de processor gegevens kan lezen en schrijven, en ROM, alleen leesbaar. De 68000 processor kan maximaal 16 megabyte RAM en/of ROM geheugen adresseren. Op een Amiga 500 zit standaard 1/2 megabyte RAM, oftewel 512 kilobyte, de rest kan via geheugen-kaarten erbij gevoegd worden. Een populaire uitbreiding is de A501 kaart; deze kan in iedere Amiga 500 gezet worden, en bevat 512k RAM. Hiermee komt het totale geheugen op 1 megabyte. En dat lijkt tegenwoordig een 'must' te worden, afgaande op de programma's die met minder dan 1 megabyte geen genoegen nemen. Voor de meeste programma's is 512k gelukkig nog meer dan genoeg. Om iedereen nog meer te verwarren: het kan ook andersom! We denken daarbij aan programma's, die extra RAM helemaal niet leuk vinden. Dit komt *door de 'custom chips', die de grafi-*

sche en muzikale mogelijkheden van de Amiga mogelijk maken. Deze chips zijn speciaal gemaakt om de 68000 te ontlasten; ze kunnen op eigen houtje gegevens uit RAM halen of erin zetten zonder hulp van de processor. Het nadeel is dat ze niet verder dan de normale 512 kbyte kunnen adresseren; daarom heet dit stuk RAM ook wel CHIP RAM dat deel van de RAM dat bestemd is voor de (custom) chips. RAM dat buiten die CHIP RAM staat, kan alleen door de 68000 gelezen en beschreven worden. Omdat de custom chips hier niet 'in de weg' zitten, is het gebruik van dit RAM iets sneller dan bij CHIP-RAM. Daarom wordt het FAST RAM genoemd. Dus, gegevens die in FAST RAM staan worden niet herkend door de custom chips. Het is de taak van de programmeur om te zorgen dat data, bestemd voor de custom chips, in CHIP-RAM terecht komt! Gebeurt dat niet, dan is het resultaat: troep op je scherm, of allerlei rare geluidjes. En er zijn nogal wat slecht gemaakte programma's, die op hun bek gaan als ze in FAST RAM terecht komen.

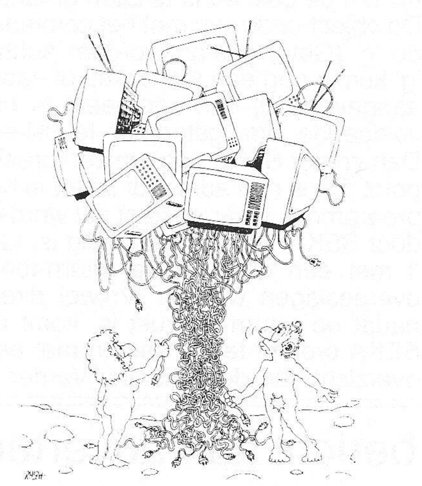
K-SEKA

Om dit soort problemen te voorkomen, raden we aan om een assembler te gebruiken, die zowel met CHIP als met FAST RAM om kan gaan. Onze keus is gevallen op de K-SEKA assembler van Kuma Computers Ltd. "Waarom SEKA?", zul je je afvragen. Er zijn meer assemblers te koop, of gratis verkrijgbaar in het Public Domain circuit, maar die hebben een paar nadelen:

- de brontekst (listing) moeten via een aparte editor geschreven worden;
- een assembler maakt van de brontekst een machinetaalprogramma (object code);
- een linker zorgt ervoor dat het een werkend programma (object program) wordt.

De SEKA assembler heeft al deze functies bij elkaar staan; er hoeft niks van disk bijgeladen te worden. Assembleren en linken gebeurt automatisch, en alle files, brontekst, machinetaal en object code, blijven in RAM staan. Dus geen laden en saven van en naar disk. Dit maakt SEKA een bijzonder snelle jongen; assembleren & linken gaat tot 25000 regels brontekst per minuut! Dat is althans wat de handleiding beweert. We hebben dat niet kunnen testen (een brontekst van 25000 regels was niet zo snel voorhanden), maar wel een programma van 1000 regels. Dat is al aardig wat, al zeggen we het zelf. Het assembleren en linken duurde iets minder dan 3 seconden. Omgerekend komt dat op ruim 20000 regels per minuut: een snellere assembler moeten wij nog tegenkomen!

Let wel: er zijn nogal wat oude versies van SEKA in gebruik, die object-code in FAST RAM zetten. Dat geeft dus wel eens probleempjes voor de trotse bezitters van geheugen uitbreidingen, de oorzaak daarvan mag nu wel bekend zijn. Er is een vernieuwde, helaas onofficiele versie in omloop, die een zelfgemaakt programma naar keuze in CHIP of FAST RAM plaatst. Het is natuurlijk overbodig om te zeggen dat dit de voorkeur verdient.



Eerste voorbeeld

Na deze inleiding wordt het tijd om eens echt met het machinetaalgebeuren te beginnen. Wie SEKA niet bezit, hoeft niet te wanhopen: voor het eerste voorbeeld kan iedere assembler gebruikt worden! Alleen is de bediening anders dan hier beschreven wordt. Raadpleeg de handleiding die bij je assembler zit, als je er niet uitkomt. Enfin, aan het werk nu! Start SEKA op vanaf CLI of workbench. We houden het nog eenvoudig, dus de workspace kan op 10 of zo gezet worden. Bezitters van de nieuwe SEKA-versie moeten nog aangeven of ze in CHIP of FAST RAM willen werken. Als je in de command-mode zit, bij de SEKA prompt, druk dan op de Esc-toets om naar de editor te gaan. Snel even het volgende programmaatje intikken:

LISTING 1

```
start:  move.l $4,a6
        lea dosnaam,a1
        jsr -408(a6)
        move.l d0,a6
        jsr -60(a6)
        move.l d0,d1
        move.l #tekst,d2
        move.l #eind-tekst,d3
        jsr -48(a6)
        clr d0
        rts
even
dosnaam: dc.b 'dos.library',0
even
tekst: dc.b 'Hallo wereld!',10
eind:
```

P.S. de regelnummers zijn hier voor het gemak. Die mag je niet intikken; ook niet bij andere assemblers!

Druk vervolgens op Esc om weer naar de command-mode te gaan. Het eerste wat we nu doen, is de brontekst saven, met het 'w' (Write) commando. **Maak hier een gewoonte van**, want zeker is zeker. Een klein foutje in de machinetaal-listing kan later een 'crash' tot gevolg hebben, en dan ben je alles KWIJT wat nog niet op disk staat! Dus: saven, zolang het nog kan. Goed, we zijn er nog niet, want het programma moet nog geassembleerd worden. Waar de meeste assemblers nu een assemble- en een link-opdracht nodig hebben, hoeft de SEKA-bezitter slechts het commando 'a' (Assemble) te geven. Er wordt naar OPTIONS gevraagd, maar die zijn op dit moment niet belangrijk. Daar kun je rustig met de Return-toets op antwoorden. Onwaarschijnlijk snel wordt alles geassembleerd, en klaar is Kees! Tenminste, als de melding "No Errors" verschijnt! Bij een fout, een typefoutje bijvoorbeeld, komt de 'defecte' regel in beeld. Dan is een snelle sprong naar de editor voldoende om de fout te kunnen herstellen. Vervolgens komen we bij het commando 'h' (Howbig) om te kijken hoe groot de verschillende files zijn geworden:

Work - plaats en grootte van de workspace;
Src - plaats en grootte van brontekst;
RelC - informatie om code later in te kunnen laden;
RelD - idem **Code** - plaats en grootte van de object-code.

Wie geen gekke dingen heeft gedaan, krijgt een overzicht die er ongeveer als volgt uitziet:

```
Work 017E70 01A670  10240
Src  017E76 017FB4    318
RelC 0181E8 0181F8     16
RelD 0181FC 018204      8
Code 018214 018258     68
```

De adressen zullen waarschijnlijk anders zijn, maar dat is niet erg. Zolang de grootte klopt, is alles in orde. Op dit moment heeft SEKA de brontekst en de assembler-code tegelijk in het geheugen zitten. We hebben al gezien dat de brontekst op disk gezet kan worden met het 'w' commando. Om

de object-code te saven, is het 'wi' commando (Write Image) nodig. Als je het programma echter later vanuit de CLI wilt opstarten, moet je het saven met het commando 'wo' (Write Object). Als je een van deze twee mogelijkheden gebruikt, wil SEKA natuurlijk weten wat er nou precies op disk gezet moet worden. Je moet daarvoor het begin- en eindadres opgeven van de object-code. Die adressen kun je met het 'h' commando opvragen. In plaats daarvan kunnen ook labels gebruikt worden die in de brontekst voorkomen. Omdat waarschijnlijk niet iedereen al eerder met een label-assembler gewerkt heeft, zullen we dit fenomeen even wat nader toelichten.

Labels

Labels zijn een soort variabelen. Ze krijgen hun waarde als het programma geassembleerd wordt. Elk label stelt namelijk een adres voor. Als we het voorbeeldprogramma bekijken, zijn dat er vier. Namelijk de labels: "start", "dosnaam", "tekst" en "eind". Tijdens het assembleren krijgen ze hun eigen (adres)waarde. De waarde van "start" wordt het beginadres van het programma. In het label "tekst" komt het adres te staan waar het zinnetje "Hallo wereld !" begint. Door die variabele te gebruiken, zie regel 7 en 8, kan binnen het programma bepaald worden waar de tekst zich bevindt. Om op het saven terug te komen: begin- en eind-adres kun je dus ook aangeven met labels; in ons geval door de woorden "start" en "eind" in te tikken. Als je de object-code op disk zet, wordt -bij de nieuwe SEKA versie- ook nog naar MODE gevraagd. Hier moet een C (CHIP) of F (FAST) ingevoerd worden. Dit bepaalt in welke soort RAM het programma later wordt geladen.

Let's go

Na al deze save-perikelen wordt het tijd om de boel eens te laten draaien. De object-code kan met het commando 'g' (Goto) gestart worden. Achter 'g' kun je nog een startadres of -label aangeven. Bij ons voorbeeld is het voldoende om 'gstart' in te tikken. Dan vraagt SEKA nog om een breakpoint. Dit is een adres (of label) in het programma, waar gestopt zal worden door SEKA. Als dat niet nodig is, kan 't met een druk op de return-toets overgeslagen worden. Vrijwel direct nadat op return gedrukt is, komt de SEKA prompt terug, samen met een overzicht van de registers. Verder is er niks gebeurt... denk je! Het voorbeeld-programma stuurde een zinnetje naar de standaard output. Dat is op dat moment het CLI-venster. Klik het SEKA-venster maar eens naar achteren, met het 2e gadget rechtsboven. Dan zie je het CLI-venster, met daarin. kijk zelf maar! Als je het object-programma op disk gezet hebt, met commando 'wo', dan kun je die vanuit de CLI opstarten. Gooi SEKA dan eventjes eruit (via '!') en tik de filenaam in die je bij het saven gebruikt hebt. Ziedaar een wonder gebeuren, en ben je gelukkig!

Tot slot

Aangezien er niets kapot kan gaan, is het prima om wat met het programma te 'rotzooien'. Probeer verschillende teksten uit, er kunnen leuke 'menuutjes' mee gemaakt worden, voeg er wat control-karakters in, b.v. ctrl-l maakt het scherm schoon. Ach, laat je fantasie de vrije loop. De volgende keer zullen we aan de hand van dit voorbeeld-programma dieper op de machinetaal-instructies ingaan. Dan bekijken we ook hoe de Amiga met zijn geheugen omgaat.
Stay tuned!

JOHAN & JOHAN